



# НТО

**МАТЕРИАЛЫ ЗАДАНИЙ**  
Всероссийской междисциплинарной олимпиады  
школьников 8–11 класса  
«Национальная технологическая олимпиада»  
по профилю  
«Разработка компьютерных игр»

(клuster «Виртуальные миры: разработка компьютерных игр, технологии виртуальной реальности, технологии дополненной реальности»)

2024/25 учебный год

*ntcontest.ru*

УДК 373.5.016:[794:004.946]

ББК 74.263.2

Р17

Авторы:

С. Е. Бойченко, И. А. Воронцов, А. А. Гаврилюк, Е. Н. Горечин, А. В. Горошкина, О. В. Зубков, Д. С. Исаев, Д. Е. Кабаева, Н. С. Лакин, М. Н. Лутфуллин, И. Б. Мамай, А. В. Резников, Е. В. Сечинский, В. Е. Ярмонов

**P17** Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада». Учебно-методическое пособие

**Том 17 Разработка компьютерных игр**

— М.: Ассоциация участников технологических кружков, 2025. — 221 с.

ISBN 978-5-908021-16-6

Данное пособие разработано коллективом авторов на основе опыта проведения всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада» в 2024/25 учебном году, а также многолетнего опыта проведения инженерных соревнований для школьников. В пособии собраны основные материалы, необходимые как для подготовки к олимпиаде, так и для углубления знаний и приобретения навыков решения инженерных задач.

В издании приведены варианты заданий по профилю Национальной технологической олимпиады за 2024/25 учебный год с ответами, подробными решениями и комментариями. Пособие адресовано учащимся 8–11 классов, абитуриентам, школьным учителям, наставникам и преподавателям учреждений дополнительного образования, центров молодежного и инновационного творчества и детских технопарков.

Методические материалы также могут быть полезны студентам и преподавателям направлений, относящихся к группам:

01.00.00 Математика и механика

02.00.00 Компьютерные и информационные науки

09.00.00 Информатика и вычислительная техника

10.00.00 Информационная безопасность

ISBN 978-5-908021-16-6

УДК 373.5.016:[794:004.946]

ББК 74.263.2



9 785908 021166 >

# Оглавление

<b>1 Введение</b>	<b>5</b>
<b>1.1 Национальная технологическая олимпиада</b>	<b>5</b>
<b>1.2 Разработка компьютерных игр</b>	<b>13</b>
<b>2 Первый отборочный этап</b>	<b>15</b>
<b>2.1 Работа наставника НТО на этапе</b>	<b>15</b>
<b>2.2 Предметный тур. Информатика</b>	<b>16</b>
2.2.1 Первая волна. Задачи 8–11 класса . . . . .	16
2.2.2 Вторая волна. Задачи 8–11 класса . . . . .	26
2.2.3 Третья волна. Задачи 8–11 класса . . . . .	36
2.2.4 Четвертая волна. Задачи 8–11 класса . . . . .	49
<b>2.3 Предметный тур. Математика</b>	<b>64</b>
2.3.1 Первая волна. Задачи 8–9 класса . . . . .	64
2.3.2 Первая волна. Задачи 10–11 класса . . . . .	67
2.3.3 Вторая волна. Задачи 8–9 класса . . . . .	71
2.3.4 Вторая волна. Задачи 10–11 класса . . . . .	74
2.3.5 Третья волна. Задачи 8–9 класса . . . . .	79
2.3.6 Третья волна. Задачи 10–11 класса . . . . .	84
2.3.7 Четвертая волна. Задачи 8–9 класса . . . . .	88
2.3.8 Четвертая волна. Задачи 10–11 класса . . . . .	92
<b>2.4 Инженерный тур</b>	<b>97</b>
<b>3 Второй отборочный этап</b>	<b>113</b>
<b>3.1 Работа наставника НТО на этапе</b>	<b>113</b>
<b>3.2 Инженерный тур</b>	<b>115</b>
3.2.1 Индивидуальные задачи . . . . .	115
3.2.2 Командные задачи . . . . .	125

---

<b>4 Заключительный этап</b>	<b>152</b>
<b>4.1 Работа наставника НТО при подготовке к этапу</b>	<b>152</b>
<b>4.2 Предметный тур</b>	<b>154</b>
4.2.1 Информатика. 8–11 классы . . . . .	154
4.2.2 Математика. 8–9 классы . . . . .	167
4.2.3 Математика. 10–11 классы . . . . .	172
<b>4.3 Инженерный тур</b>	<b>177</b>
4.3.1 Общая информация . . . . .	177
4.3.2 Легенда задачи . . . . .	177
4.3.3 Требования к команде и компетенциям участников . . . . .	178
4.3.4 Оборудование и программное обеспечение . . . . .	178
4.3.5 Описание задачи . . . . .	179
4.3.6 Система оценивания . . . . .	199
4.3.7 Материалы для подготовки . . . . .	217
<b>5 Критерии определения победителей и призеров</b>	<b>219</b>
<b>6 Работа наставника после НТО</b>	<b>221</b>

# **1. Введение**

## **1.1. Национальная технологическая олимпиада**

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» (далее — Олимпиада, НТО) проводится в соответствии с распоряжением Правительства Российской Федерации от 10.02.2022 № 211-р при координации Министерства науки и высшего образования Российской Федерации и при содействии Министерства просвещения Российской Федерации, Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, Министерства промышленности и торговли Российской Федерации, Ассоциации участников технологических кружков, Агентства стратегических инициатив по продвижению новых проектов, АНО «Россия — страна возможностей», АНО «Платформа Национальной технологической инициативы» и Российского движения детей и молодежи «Движение Первых».

Проектное управление Олимпиадой осуществляет структурное подразделение Национального исследовательского университета «Высшая школа экономики» — Центр Национальной технологической олимпиады. Организационный комитет по подготовке и проведению Национальной технологической олимпиады возглавляют первый заместитель Руководителя Администрации Президента Российской Федерации С. В. Кириенко и заместитель Председателя Правительства Российской Федерации Д. Н. Чернышенко.

Национальная технологическая олимпиада — это командная инженерная Олимпиада, позволяющая школьникам работать в самых передовых инженерных направлениях. Она базируется на опыте Олимпиады Кружкового движения НТИ и проводится с 2015 года, а с 2016 года входит в перечень Российского совета олимпиад школьников и дает победителям и призерам льготы при поступлении в университеты.

Всего заявки на участие в десятом юбилейном сезоне (2024–25 гг.) самых масштабных в России командных инженерных соревнованиях подали более 140 тысяч школьников. Общий охват олимпиады с 2015 года превысил 880 тысяч участников.

НТО способствует формированию профессиональной траектории школьников, увлеченных научно-техническим творчеством и помогает им:

- определить свой интерес в мире современных технологий;
- получить опыт решения комплексных инженерных задач;
- осознанно выбрать вуз для продолжения обучения и поступить в него на льготных условиях.

Кроме того, НТО позволяет каждому участнику познакомиться с перспективными направлениями технологического развития, ведущими экспертами и найти единомышленников.

## ***Ценности НТО***

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Олимпиада создает уникальное пространство, основанное на общих ценностях и смыслах, которыми делятся все участники процесса: школьники, студенты, организаторы, наставники и эксперты. В основе Олимпиады лежит представление о современном технологическом образовании как новом укладе жизни в быстро меняющемся мире. Эта модель предполагает:

- доступность качественного обучения для всех, кто стремится к знаниям;
- возможность непрерывного развития;
- совместное формирование среды, где гуманитарные знания и новые технологии взаимно усиливают друг друга.

Это — образ общества будущего, в котором участники Олимпиады оказываются уже сегодня.

### ***Решать прикладные задачи, нацеленные на умножение общественного блага***

В заданиях Олимпиады используются актуальные вызовы науки и технологий, адаптированные под уровень школьников. Они имеют прикладной характер и отражают реальные потребности общества, а системное и профессиональное решение подобных задач способствует развитию общего блага. Олимпиада предоставляет возможность попробовать себя в этом направлении уже сегодня и найти единомышленников.

### ***Создавать, а не только потреблять***

Стремление к созданию нового ценится выше потребления готового, а ориентация на общественную пользу — выше личной выгоды. Это не исключает заботу о собственных интересах, но подчеркивает: творчество приносит больше удовлетворения, чем пассивное потребление. Олимпиада — совместный труд организаторов, партнеров и участников, в котором важнее стремление решать общие задачи, чем критика чужих усилий.

### ***Работать в команде***

Командная работа рассматривается не только как эффективный способ достижения целей, но и как основа для формирования сообщества, объединенного общими ценностями. Команда помогает раскрыть индивидуальность каждого, при этом сохраняя уважение к другим. Такие горизонтальные связи необходимы для реализации амбициозных технологических проектов. Олимпиада способствует формированию подобного сообщества и приглашает к его созданию всех заинтересованных.

### ***Осваивать и ответственно развивать новые технологии***

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ, объединенные интересом к современным технологиям, стремлением

---

к их пониманию и созданию нового. Возможности технологий постоянно расширяются, однако развитие должно сопровождаться ответственностью. Этика инженера и ученого предполагает осознание последствий своих решений. Главное правило — создавая новое, не навредить.

### ***ИграТЬ честно и пробовАть себя***

Ценится честная победа, достигнутая в рамках установленных правил. Это предполагает отказ от списывания, давления и манипуляций. Честная игра означает уважение к себе, команде и соперникам. Олимпиада поддерживается как безопасное пространство, где каждый может пробовать новое, не опасаясь ошибок, и постепенно становиться сильнее и увереннее в себе.

### ***Быть человеком***

Соревнования — это сложный и эмоционально насыщенный процесс, в котором особенно важны порядочность, вежливость и чуткость. Эмпатия, уважение и забота делают участие полезным и комфортным. Высоко ценится бережное отношение к людям и их труду, отказ от токсичной критики и готовность нести ответственность за слова и поступки. Участие в общем деле помогает не только окружающим, но и самому человеку.

## ***Организационная структура НТО***

НТО — межпредметная олимпиада. Спектр соревновательных направлений (профилей НТО) сформирован на основе актуального технологического пакета и связан с решением современных проблем в различных технологических отраслях. С полным перечнем направлений (профилей) можно ознакомиться на сайте НТО: <https://ntcontest.ru/tracks/nto-school/>.

Соревнования в рамках НТО проводятся по четырем трекам:

1. НТО Junior для школьников (5–7 классы).
2. НТО школьников (8–11 классы).
3. НТО студентов.
4. Конкурс цифровых портфолио «Талант НТО».

В 2024/25 учебном году 21 профиль НТО включен в Перечень олимпиад школьников, ежегодно утверждаемый Приказом Министерства науки и высшего образования Российской Федерации, а также в Перечень олимпиад и иных интеллектуальных и (или) творческих конкурсов, утверждаемый приказом Министерства просвещения Российской Федерации. Это дает право победителям и призерам профилей НТО поступать в вузы страны без вступительных испытаний (БВИ), получить 100 баллов ЕГЭ или дополнительные 10 баллов за индивидуальные достижения. Преимущества при поступлении победителям и призерам НТО предлагают более 100 российских вузов.

НТО для школьников 8–11 классов проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. Участникам предлагаются предметный тур, состоящий из задач по двум предметам, связанным

с выбранным профилем, а также инженерный тур, задания которого погружают участников в тематику профиля; образовательный модуль формирует теоретические знания и представления.

- Второй отборочный этап — заочный командный. На этом этапе участники выполняют как индивидуальные задания на проверку компетенций, так и командные задачи, соответствующие выбранному профилю.
- Заключительный этап — очный командный. В течение 5–6 дней команды участников со всей страны, успешно прошедшие оба отборочных этапа, соревнуются в решении комплексных прикладных инженерных задач.

### ***Профили НТО 2024/25 учебного года и соответствующий уровень РСОШ***

#### **Профили II уровня РСОШ:**

- Автоматизация бизнес-процессов.
- Автономные транспортные системы.
- Беспилотные авиационные системы.
- Водные робототехнические системы.
- Инженерные биологические системы.
- Наносистемы и наноинженерия.
- Нейротехнологии и когнитивные науки.
- Технологии беспроводной связи.
- Цифровые технологии в архитектуре.
- Ядерные технологии.

#### **Профили III уровня РСОШ:**

- Анализ космических снимков и геопространственных данных.
- Аэрокосмические системы.
- Большие данные и машинное обучение.
- Геномное редактирование.
- Интеллектуальные робототехнические системы.
- Интеллектуальные энергетические системы.
- Информационная безопасность.
- Искусственный интеллект.
- Летающая робототехника.
- Спутниковые системы.
- Кластер «Виртуальные миры»:
  - ◊ Разработка компьютерных игр.
  - ◊ Технологии виртуальной реальности.
  - ◊ Технологии дополненной реальности.

#### **Профили без уровня РСОШ:**

- Инфохимия.
- Квантовый инжиниринг.
- Новые материалы.
- Программная инженерия в финансовых технологиях.

- 
- Современная пищевая инженерия.
  - Умный город.
  - Урбанистика.
  - Цифровые сенсорные системы.
  - Разработка мобильных приложений.

Обратите внимание на то, что в олимпиаде 2025/26 учебного года список профилей, в т. ч. входящих в РСОШ, и уровни РСОШ могут поменяться.

Участие в НТО старшеклассников может принять любой школьник, обучающийся в 8–11 классе. Чаще всего Олимпиада привлекает:

- учащихся технологических кружков, интересующихся инженерными и роботехническими соревнованиями;
- школьников, увлеченных олимпиадами и предпочитающих межпредметный подход;
- энтузиастов передовых технологий;
- активных участников хакатонов, проектных конкурсов и профильных школ;
- будущих предпринимателей, ищащих команду для реализации стартап-идей;
- любознательных школьников, стремящихся выйти за рамки школьной программы.

Познакомить школьников с НТО и ее направлениями, а также мотивировать их на участие в Олимпиаде можно с помощью специальных мероприятий — Урока НТО и Дней НТО. Методические рекомендации для педагогов по проведению Урока НТО и организации Дня НТО в образовательной организации размещены на сайте: <https://nti-lesson.ru>. Здесь можно подобрать и скачать готовые сценарии занятий и подборки материалов по различным направлениям Олимпиады.

Участвуя в НТО, школьники получают возможность работать с практико-ориентированными задачами в области прорывных технологий, собирать команды единомышленников, погружаться в профессиональное сообщество, а также заработать льготы для поступления в вузы.

По всей стране работают площадки подготовки к НТО, которые помогают привлекать участников и проводят мероприятия по подготовке к этапам Олимпиады. Такие площадки могут быть открыты на базе:

- школ и учреждений дополнительного образования;
- частных кружков по программированию, робототехнике и другим технологическим направлениям;
- вузов;
- технопарков и других образовательных и научно-технических организаций.

Любое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к Олимпиаде и присоединиться к Кружковому движению НТИ. Подробные инструкции о том, как стать площадкой подготовки, размещены на сайте: <https://ntcontest.ru>. Условия регистрации и требования к ним актуализируются с развитием Олимпиады, а обновленная информация публикуется перед началом каждого нового цикла.

## Наставники НТО

В Национальной технологической олимпиаде большое внимание уделяется работе с **наставниками** — людьми, сопровождающими участников на всех этапах подготовки и участия в Олимпиаде. Наставник оказывает поддержку как в решении организационных вопросов, так и в развитии технических и социальных навыков школьников, включая умение работать в команде.

Наставником НТО может стать любой взрослый, готовый помогать школьникам развиваться и готовиться к участию в инженерных соревнованиях. Это может быть:

- учитель школы или преподаватель вуза;
- педагог дополнительного образования;
- руководитель кружка;
- родитель школьника;
- специалист из технологической области или представитель бизнеса.

Даже если наставник сам не обладает достаточными знаниями в определенной области, он может привлекать к подготовке коллег и экспертов, а также оказывать поддержку и организовывать процесс обучения для самостоятельных учеников. Сегодня сообщество наставников НТО насчитывает более **7 000 человек** по всей стране.

Главная цель наставника — **организовать системную подготовку к Олимпиаде в течение всего учебного года**, поддерживать интерес и мотивацию участников, а также помочь им справляться с возникающими трудностями. Также наставник фиксирует цели команды и каждого участника, чтобы в дальнейшем можно было проанализировать развитие профессиональных и личных компетенций.

### *Основные направления работы наставника*

Организационные задачи:

- Информирование и мотивация: наставник рассказывает учащимся об НТО, ее этапах и преимуществах, помогает с выбором подходящего профиля, ориентируясь на интересы и способности школьников.
- Составление программы подготовки: формируется расписание и план занятий, организуется работа по освоению необходимых знаний и навыков.
- Контроль сроков: наставник следит за календарем Олимпиады и напоминает участникам о сроках решения заданий отборочных этапов.

Содержательная подготовка:

- Оценка компетенций участников: наставник помогает определить сильные и слабые стороны учеников и подбирает задания и материалы для устранения пробелов.
- Подготовка к отборочным этапам: помочь в изучении рекомендованных материалов, заданий прошлых лет, онлайн-курсы по профилям.
- Подготовка к заключительному этапу: разбираются задачи заключительных этапов прошлых лет, отслеживаются подготовительные мероприятия (очные и дистанционные), в которых наставник рекомендует ученикам участвовать.

Развитие личных и командных навыков:

- Формирование команд: наставник помогает сформировать сбалансированные команды для второго отборочного и финального этапов, распределить роли, при необходимости ищет участников из других регионов и организует онлайн-коммуникацию.
- Анализ прогресса и опыта: после каждого этапа проводится совместная рефлексия, обсуждаются успехи и трудности, выявляются зоны роста и направления для дальнейшего развития.
- Поддержка и мотивация: наставник поддерживает интерес и энтузиазм участников (особенно в случае неудачных результатов), помогает справиться с разочарованием и сохранить настрой на дальнейшее участие.
- Построение индивидуальной образовательной траектории: наставник помогает школьникам осознанно планировать дальнейшее обучение: выбирать курсы, участвовать в конкурсах, определяться с вузами и направлениями подготовки.

## **Поддержка наставников НТО**

Работе наставников посвящен отдельный раздел на сайте НТО: <https://ntcontest.ru/mentors/>.

Для систематизации знаний и подходов к работе наставников в рамках инженерных соревнований разработан курс «Дао начинающего наставника: как сопровождать инженерные команды»: <https://stepik.org/course/124633/>. Курс формирует общие представления об их работе в области подготовки участников к инженерным соревнованиям.

Для совершенствования профессиональных компетенций по направлениям профилей создан курс «Дао начинающего наставника: как развивать технологические компетенции»: <https://stepik.org/course/186928/>.

Для организации занятий с учениками педагогам предлагаются образовательные программы, разработанные на основе многолетнего опыта организации подготовки к НТО. В настоящий момент они представлены по передовым технологическим направлениям:

- компьютерное зрение;
- геномное редактирование;
- водная, летающая и интеллектуальная робототехника;
- машинное обучение и искусственный интеллект;
- нейротехнологии;
- беспроводная связь, дополненная реальность.

Программы доступны на сайте: <https://ntcontest.ru/mentors/education-programs/>.

Регистрируясь на платформе НТО, наставники получают доступ к личному кабинету, в котором отображается расписание отборочных соревнований и мероприятий по подготовке, требования к знаниям и компетенциям при решении задач отборочных этапов.

Сообщество наставников НТО существует и развивается. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: <https://konkurs.kruzhok.org/>. Принять участие в конкурсе может каждый наставник.

В 2022 году было выпущено пособие «Технологическая подготовка инженерных команд. Методические рекомендации для наставников». Методические рекомендации предназначены для учителей технологий, а также наставников и педагогов кружков и центров дополнительного образования. Рекомендации направлены на помочь в процессе преподавания технологий в школе или в кружке. Пособие построено на примерах из реального опыта работы со школьниками, состоит из теоретических положений, посвященных популярным взглядам в педагогике на тему подготовки инженерных команд к соревнованиям. Электронное издание доступно по ссылке: <https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih>.

В нем рассмотрены особенности подготовки к пяти направлениям:

- Большие данные.
- Машинное обучение.
- Искусственный интеллект.
- Спутниковые системы.
- Летающая робототехника.

Для наставников НТО разработана и постоянно пополняется страница с материалами для профессионального развития: <https://nto-forever.notion.site/c9b9cbd21542479b97a3fa562d15e32a>.

## **1.2. Разработка компьютерных игр**

С апреля 2024 года в рамках Национальной технологической олимпиады начал работу новый укрупненный профиль Виртуальные миры, объединивший три ранее самостоятельных направления: Разработка компьютерных игр, Технологии виртуальной реальности и Технологии дополненной реальности. Такое объединение позволяет усилить междисциплинарность заданий, расширить возможности для развития участников и сформировать более целостный образовательный трек.

Подпрофиль Разработка компьютерных игр ориентирован на решение прикладных технологических задач, поставленных представителями индустрии. В его основе — интеграция теоретических знаний и практических навыков, командный подход с четким разделением ролей, развитие проектов с возможностью их последующей доработки и коммерциализации.

Работа в рамках подпрофиля включает использование современных цифровых инструментов и технологий:

- сред разработки;
- искусственного интеллекта;
- анализа пользовательских данных.

Участники проектируют и создают прототипы игровых продуктов по заданной теме, встраивая в них актуальные механики и решения. В качестве сквозных технологий подпрофиля можно выделить:

- технологий хранения и анализа больших данных;
- искусственный интеллект;
- нейротехнологии.

Решение задач, поставленных перед участниками, требует серьезной подготовки в области математики, информатики, цифрового творчества и логического мышления.

Олимпиада строится как трехэтапная траектория: два отборочных этапа (индивидуальный и командный) и заключительный, проектный. Итогом работы становится командная разработка игрового прототипа, соответствующего индустриальным стандартам и обладающего потенциалом дальнейшего развития.

В сезоне 2024–25 гг. задачей второго и заключительного этапа вновь стало создание игры в рамках платформы «Берлога» — образовательного пространства, объединяющего серию мобильных игр с единым сюжетом. «Берлога» служит точкой входа в технологическое образование: через увлекательные игровые механики она вовлекает школьников в освоение программирования, в участие в инженерных соревнованиях и развитие цифровых навыков.

Тематика заданий выстраивается через систему логически связанных этапов: в отборочных турах участники формируют необходимые для финала компетенции. Все блоки — «Программирование», «Геймдизайн», «Арт», «Продукт» — направлены на развитие как индивидуальных, так и командных навыков. На втором этапе именно взаимодействие ролей внутри команды становится ключом к достижению результата в установленные сроки.

---

Для погружения в профессиональный контекст организуются лекции и мастер-классы от представителей индустрии. Эксперты рассказывают об актуальных технологиях, подходах к разработке, возможностях трудоустройства и развитии в сфере игр. Это расширяет представление школьников о будущей карьере и дает им практическую ориентировку в индустрии.

Примеры карьерного трека выпускников профиля:

- Финалист ОНТИ GameDev (2019–2020 гг.) и победитель GameNet Challenge разработал гиперказуальную игру, получил предложение о сотрудничестве от Ducky LTD, в настоящий момент он работает геймдизайнером и принимает участие в разработке заданий профиля.
- Победители ОНТИ GameDev (2020–2021 гг.) и финалисты GameNet Challenge основали игровую студию, активно развивают портфолио, участвуют в международных Game Jam, готовятся к работе с индустриальными заказчиками.
- Победитель ОНТИ: GameDev — программист-разработчик — работает над созданием RTS-стратегии.
- Финалисты РКИ НТО 2022–2023 и 2023–2024 гг. сотрудничают с компанией Ninsar.Games, участвуют в организации профиля и менторской поддержке новых команд.
- Победители 2022–2023 гг. опубликовали игру «Как в старой сказке» на платформе VK Play в марте 2024 года. Прототип игры был разработан в рамках участия в финале НТО.
- Финалисты РКИ НТО 2023–2024 гг. сотрудничают с платформой «Берлога», дорабатывают и публикуют собственные игры.

## **2. Первый отборочный этап**

### **2.1. Работа наставника НТО на этапе**

Педагог-наставник играет важную роль в подготовке участника к первому отборочному этапу Национальной технологической олимпиады. На этом этапе школьникам предстоит справиться как с предметными задачами, соответствующими профилю, так и с заданиями инженерного тура, погружающими в выбранную технологическую область.

Наставник может организовать подготовку участника, используя разнообразные форматы и ресурсы:

- Разбор заданий прошлых лет. Совместный анализ задач отборочного этапа предыдущих лет позволяет понять структуру, уровень сложности и типичные подходы к решению. Это формирует у школьника устойчивые стратегии работы с олимпиадными заданиями.
- Мини-соревнования. Проведение тренировочных турниров с заданиями предметных олимпиад муниципального уровня помогает развить соревновательный навык, тренирует скорость и уверенность при решении задач в ограниченное время.
- Углубленные занятия. Наставник может выстроить образовательную траекторию, опираясь на рекомендации разработчиков профиля, и провести занятия по ключевым темам. Это особенно важно для системного понимания предметной области.
- Использование онлайн-курсов. Для самостоятельной подготовки и проверки знаний участник может использовать предметные курсы НТО, размещенные на платформах Степик и Яндекс Конкурс. Наставник может также организовать занятия с использованием этих материалов в рамках групповой или индивидуальной подготовки.
- Привлечение внешних экспертов. Если у наставника нет достаточной экспертизы в какой-либо предметной области, он может пригласить других педагогов или специалистов для проведения тематических занятий.
- Поддержка в инженерном туре. Инженерный тур включает теоретические материалы и задания, помогающие глубже погрузиться в тематику профиля. Наставник может сопровождать изучение курса, помогать в разборе теоретических вопросов и тренировать участника на практических задачах.

Таким образом, наставник не только помогает систематизировать подготовку, но и мотивирует участника, создавая для него комфортную и продуктивную образовательную среду.

## 2.2. Предметный тур. Информатика

### 2.2.1. Первая волна. Задачи 8–11 класса

Задачи первой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63452/enter/>.

#### **Задача 2.2.1.1. Ускорение ускорения (10 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Рассмотрим модель движения тела. Будем фиксировать такие параметры, как координата, скорость, ускорение и ускорение ускорения (рывок). Если некоторый параметр равен  $a$  и имеет скорость изменения  $v$ , то в следующий момент времени этот параметр будет равен  $a + v$ .

Например, если тело имело координату, равную 10, скорость, равную 20, ускорение, равное 30 и ускорение ускорения, равное 40, то в следующий момент оно будет иметь координату 30, скорость 50 и ускорение 70. Ускорение ускорения будем считать в этой задаче постоянной величиной.

Задача довольно проста: тело в начальный момент времени 0 находится в точке с координатой 0, скоростью 0 и ускорением 0. На это тело действует постоянное ускорение ускорения, равное 6. Требуется определить, в точке с какой координатой окажется это тело в момент времени  $t$ .

#### **Формат входных данных**

В единственной строке находится одно число  $t$ , где  $0 \leq t \leq 10^6$ .

#### **Формат выходных данных**

Вывести одно число — координату, в которой окажется тело в момент времени  $t$ .

## Примеры

*Пример №1*

<b>Стандартный ввод</b>
6
<b>Стандартный вывод</b>
120

*Пример №2*

<b>Стандартный ввод</b>
2
<b>Стандартный вывод</b>
0

*Пример №3*

<b>Стандартный ввод</b>
1000000
<b>Стандартный вывод</b>
999997000002000000

## Решение

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int t;
6     cin >> t;
7     cout << ((t * (t - 1)) * (t - 2)) << endl;
8 }
```

## Задача 2.2.1.2. Двойное остекление (15 баллов)

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

## **Условие**

У деда Василия есть два прямоугольных куска стекла. Один из них имеет размеры  $a \times b$ , другой —  $c \times d$ . Дед собирается из этих кусков сделать окно с двойным остеклением. Он хочет, чтобы окно было обязательно квадратным и как можно большим по размеру. Дед должен вырезать из имеющихся у него прямоугольников два одинаковых квадрата максимально возможного размера. Нужно написать программу, которая по заданным  $a, b, c, d$  найдет максимальные размеры квадратного окна. Имейте ввиду, что оба квадрата могут быть вырезаны и из одного прямоугольного куска стекла.

## **Формат входных данных**

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника  $a, b$  через пробел, во второй — размеры второго прямоугольника  $c, d$  через пробел, где  $1 \leq a, b, c, d \leq 10^9$ .

## **Формат выходных данных**

Вывести одно число — максимальную сторону квадратного двойного окна, которое можно вырезать из заданных на входе прямоугольных кусков стекла. Ответ может быть нецелым, требуется вывести его с точностью 1 знак после десятичной точки.

## **Примеры**

### *Пример №1*

<b>Стандартный ввод</b>
5 10
9 6
<b>Стандартный вывод</b>
5

### *Пример №2*

<b>Стандартный ввод</b>
4 10
9 6
<b>Стандартный вывод</b>
4 . 5

## **Комментарий**

Второй пример показывает, что иногда лучше вырезать оба квадрата из одного и того же куска стекла.

## **Решение**

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     double a, b, c, d;
6     cin >> a >> b >> c >> d;
7     double a0 = min({a, b, c, d});
8     double a1 = min(max(a, b) / 2.0, min(a, b));
9     double a2 = min(max(c, d) / 2.0, min(c, d));
10    double ans = max({a0, a1, a2});
11    if( (int)ans == ans ){
12        int ians = ans;
13        cout << ians << endl;
14        return 0;
15    }
16    cout.precision(1);
17    cout << fixed << ans << endl;
18 }
```

## **Задача 2.2.1.3. О золотой рыбке и... досках (20 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

### **Условие**

После событий известной сказки А. С. Пушкина старик решил принципиально не пользоваться услугами золотой рыбки. Поэтому для того чтобы изготовить новое корыто, он честно заготовил  $n$  одинаковых досок.

Но гостивший в это время у старика со старухой внук решил, что ему нужно научиться пилить. И, не сказав ничего своему деду, внук быстро распилил каждую из досок на две части. В итоге у старика оказались  $2n$  кусков досок. Самое интересное, что все эти куски оказались разными по длине, но имели целочисленные размеры. К сожалению, старику забыл, какова была исходная длина целых досок.

### **Формат входных данных**

В первой строке задается целое число  $n$  — исходное количество целых досок, где  $1 \leq n \leq 10^5$ .

Во второй строке заданы  $2n$  целых чисел  $d_i$  — длины всех кусков, которые получились после «тренировки» внука, где  $1 \leq d_i \leq 10^9$ . Гарантируется, что эти числа попарно различны, и их можно разбить на пары одинаковых по сумме чисел.

Все эти части досок пронумерованы от 1 до  $2n$  в том порядке, в котором они заданы на входе.

### **Формат выходных данных**

В первую строку вывести одно число — исходную длину целых досок.

В следующих  $n$  строках вывести пары номеров кусков досок, которые составляют по длине целые доски. Номера выводить через один пробел, внутри пары сначала должен идти меньший номер, затем больший. Пары должны быть выведены в порядке возрастания первых номеров в парах.

### **Примеры**

#### *Пример №1*

Стандартный ввод
3
4 8 2 3 6 7
Стандартный вывод
10 1 5 2 3 4 6

### **Комментарий**

Отсортируем куски и далее будем брать один из начала и второй к нему из конца.

### **Решение**

Ниже представлено решение на языке C++.

#### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int n;
6     cin >> n;
7     vector<pair<int, int> > v(2 * n);
8     for(int i = 0; i < 2 * n; i++){
9         int d;
10        cin >> d;
11        v[i] = {d, i + 1};
12    }
13    sort(v.begin(), v.end());
14    vector<pair<int, int> > ans(n);
15    for(int i = 0; i < n; i++){

```

```

16     ans[i] = {v[i].second, v[2 * n - i - 1].second};
17     if(ans[i].first > ans[i].second){
18         swap(ans[i].first, ans[i].second);
19     }
20 }
21 sort(ans.begin(), ans.end());
22 cout << v[0].first + v.back().first << endl;
23 for(int i = 0; i < n; i++){
24     cout << ans[i].first << ' ' << ans[i].second << endl;
25 }
26 }
```

### **Задача 2.2.1.4. Бонусы и экономия (25 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Технология производства некоторой металлической детали предполагает вытачивание ее из металлической заготовки. При этом образуются стружки, которые не стоит выкидывать. Ведь из  $a$  комплектов стружек (оставшихся после обработки  $a$  заготовок) можно бесплатно выплавить еще одну заготовку, которую снова можно использовать для выточки детали и создания еще одного комплекта стружек.

Заготовки можно купить на оптовом складе, при этом в целях привлечения клиентов, проводится акция «купи  $b$  заготовок, тогда еще одну получишь бесплатно».

Требуется изготовить  $c$  деталей. Нужно определить минимальное число заготовок, которые нужно купить за деньги, чтобы с учетом бонусных заготовок и экономии на стружках можно было изготовить требуемое число деталей.

#### **Формат входных данных**

В одной строке через пробел заданы три целых числа  $a$ ,  $b$ , и  $c$  такие, что  $2 \leqslant a \leqslant 10^{18}$ ,  $1 \leqslant b$ ,  $c \leqslant 10^{18}$ .

#### **Формат выходных данных**

Вывести одно целое число — минимальное количество заготовок, которые нужно купить, чтобы с учетом всех бонусов и экономии выточить  $c$  конечных деталей.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
4 5 41
<b>Стандартный вывод</b>
26

## Примечания

В примере из условия нужно закупить 26 заготовок. Тогда за каждые пять купленных заготовок будет предоставлена одна бесплатная, итого по акции добавится еще пять заготовок, то есть получится 31 заготовка. Далее из 31 заготовки выточится 31 деталь, останется 31 комплект стружек. Из каждого четырех комплектов выплавится дополнительная заготовка, получится семь заготовок и три комплекта стружек. Из семи заготовок выточится семь деталей и останется семь комплектов стружек, три комплекта стружек осталось с первого шага, итого 10 комплектов стружек. Из них выплавится еще две заготовки, дающие две детали и два комплекта стружек. Собрав эти два комплекта с двумя, оставшимися от 10, получим еще одну заготовку, из которой выточится еще одна деталь. Останется один комплект стружек, который уже никак не получится использовать. Итого будет произведена  $31 + 7 + 2 + 1 = 41$  деталь.

## Комментарий

Методом бинарного поиска можно подобрать минимальное необходимое количество исходных заготовок.

## Решение

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int f1(int M, int a){
5     int res = 0, z = 0;
6     while(1){
7         if(M == 0 && z < a){
8             return res;
9         }
10        res += M;
11        M = M + z;
12        z = M % a;
13        M = M / a;
14    }
15 }
```

```

16 int f2(int M, int b){
17     return M + M / b;
18 }
19 signed main(){
20     int a, b, c;
21     cin >> a >> b >> c;
22     int L = 0, R = 1;
23     while(f1(R, a) <= c){
24         R *= 2;
25     }
26     while(R - L > 1){
27         int M = (R + L) / 2;
28         if(f1(M, a) < c){
29             L = M;
30         }
31         else{
32             R = M;
33         }
34     }
35     int z = R;
36     L = 0, R = 1;
37     while(f2(R, b) <= z){
38         R *= 2;
39     }
40     while(R - L > 1){
41         int M = (R + L) / 2;
42         if(f2(M, b) < z){
43             L = M;
44         }
45         else{
46             R = M;
47         }
48     }
49     cout << R << endl;
50 }
```

### **Задача 2.2.1.5. Сон таксиста (30 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Одному таксисту приснился красочный сон. Во сне он живет и работает в некотором городе, где абсолютно все улицы с односторонним движением. Эти улицы устроены так, что невозможно проехать с какого-либо перекрестка так, чтобы вернуться обратно на этот же перекресток, то есть в дорожной сети города нет циклов.

Таким образом, если с перекрестка  $A$  можно попасть по направлению движения улиц на перекресток  $B$ , то люди вызывают такси, иначе их везет специальный муниципальный подземный транспорт бесплатно.

В связи с такими странными правилами, таксистам в этом городе разрешено законом везти пассажира по любому маршруту, не нарушающему направления движения. Все в этом городе привыкли к такой ситуации и абсолютно спокойно относятся к тому, что таксисты везут их самым длинным путем. Разумеется, заработка таксиста за одну поездку прямо пропорционален ее длине. Для упрощения будем считать, что стоимость 1 км поездки составляет ровно 1 руб.

Схема дорог города задана. Перекрестки города пронумерованы числами от 1 до  $n$ . Таксист в своем сне находится на перекрестке номер  $S$ . Напишите программу, которая подскажет ему, сколько он максимально сможет заработать, когда ему придет заказ от клиента. Так как он не знает, куда попросит его везти клиент, нужно для каждого перекрестка от 1 до  $n$  указать максимальную стоимость поездки до этого перекрестка из пункта  $S$  на такси. Если по правилам на такси добраться из пункта  $S$  до какого-то перекрестка нельзя, вывести  $-1$ .

### **Формат входных данных**

Дорожная сеть задана следующим образом: в первой строке находятся два числа через пробел  $n$  и  $m$  — число перекрестков и число улиц в городе, где  $2 \leq n, m \leq 2 \cdot 10^5$ .

В следующих  $m$  строках задана очередная односторонняя улица в виде трех чисел  $A$ ,  $B$ ,  $d$  через пробел, где  $A$  — начало улицы,  $B$  — конец улицы и  $d$  — ее длина.  $1 \leq A, B \leq n$ ,  $1 \leq d \leq 10^9$ . Гарантируется, что в этой дорожной сети нет циклов. Некоторые пары перекрестков могут быть соединены двумя и более односторонними улицами. Дорожная сеть может быть неплоской за счет мостов и тоннелей.

В последней строке ввода содержится номер стартового перекрестка  $S$ ,  $1 \leq S \leq n$ .

### **Формат выходных данных**

Вызвести  $n$  чисел в одну строку через пробел.  $i$ -е число обозначает длину самого длинного пути с перекрестка номер  $S$  до перекрестка номер  $i$ . Если до перекрестка номер  $i$  от  $S$  нельзя доехать, не нарушая правила движения, вывести  $-1$ .

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
10 20 9 10 15 9 8 3 8 10 7 7 8 4 7 10 10 5 8 2 5 9 10

**Стандартный ввод**

```
5 6 5
7 6 5
4 6 8
3 6 4
3 4 6
5 3 2
2 5 2
2 3 3
3 1 5
1 4 2
2 1 7
4 7 4
6 8 1
5
```

**Стандартный вывод**

```
7 -1 2 9 0 18 13 19 10 26
```

**Комментарий**

Задача решается методом динамического программирования на ориентированном ациклическом графе.

**Решение**

Ниже представлено решение на языке C++.

**C++**

```
1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int n, m;
5 vector<vector<pair<int, int>>> G;
6 vector<int> order, used;
7 void dfs(int a){
8     used[a] = 1;
9     for(auto to : G[a]){
10         if(!used[to.first]){
11             dfs(to.first);
12         }
13     }
14     order.push_back(a);
15 }
16 signed main(){
17     cin >> n >> m;
18     G.resize(n + 1);
19     used.resize(n + 1, 0);
20     for(int i = 0; i < m; i++){
21         int a, b, d;
22         cin >> a >> b >> d;
23         G[a].push_back({b, d});
24     }
}
```

```

25     int s;
26     cin >> s;
27     dfs(s);
28     reverse(order.begin(), order.end());
29     vector<int> dp(n + 1, -1);
30     dp[s] = 0;
31     for(auto el : order){
32         for(auto to : G[el]){
33             dp[to.first] = max(dp[to.first], dp[el] + to.second);
34         }
35     }
36     for(int i = 1; i <= n; i++){
37         cout << dp[i] << ' ';
38     }
39 }
```

## 2.2.2. Вторая волна. Задачи 8–11 класса

Задачи второй волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63454/enter/>.

### **Задача 2.2.2.1. Игра на планшете (10 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Маленький Андрей изучает геометрические фигуры при помощи игры на планшете. У него есть прямоугольные треугольники четырех цветов и ориентаций: желтые, зеленые, красные и синие. Для каждой разновидности треугольников есть заданное количество экземпляров этих треугольников. Более точно: у Андрея есть  $a$  желтых,  $b$  зеленых,  $c$  красных и  $d$  синих треугольников. Помимо этого у него есть прямоугольная таблица  $n \times m$ .

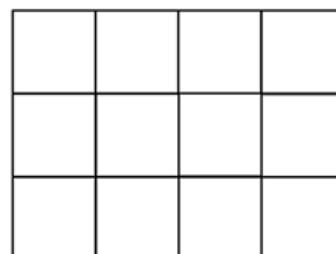
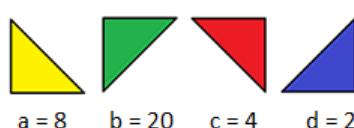


Рис. 2.2.1

Треугольники одного цвета имеют одну и ту же ориентацию, которую нельзя поменять. Андрей может только взять очередной треугольник и переместить его параллельным сдвигом в одну из ячеек этой прямоугольной таблицы. При этом в одну ячейку можно поместить либо вместе желтый и красный треугольники, либо вместе зеленый и синий, либо один любой треугольник из имеющихся.

Андрей хочет расположить в ячейках таблицы как можно больше треугольников из тех, что у него имеются. Нужно подсказать ему максимальное количество треугольников, которые получится разместить в таблице.

### ***Формат входных данных***

В первой строке содержатся четыре целых числа  $a$ ,  $b$ ,  $c$  и  $d$  через пробел — количество желтых, зеленых, красных и синих треугольников соответственно.

Во второй строке содержатся два целых числа  $n$  и  $m$  через пробел — размеры прямоугольной таблицы.

Все числа в пределах от 1 до  $10^9$ .

### ***Формат выходных данных***

Вывести одно число — максимальное количество треугольников, которые можно при заданных условиях разместить в таблице.

### ***Примеры***

#### *Пример №1*

<b>Стандартный ввод</b>
8 20 4 2
3 4
<b>Стандартный вывод</b>
18

### ***Примечания***

На рис. 2.2.2 представлен один из примеров размещения 18 треугольников из 34 заданных на входе.

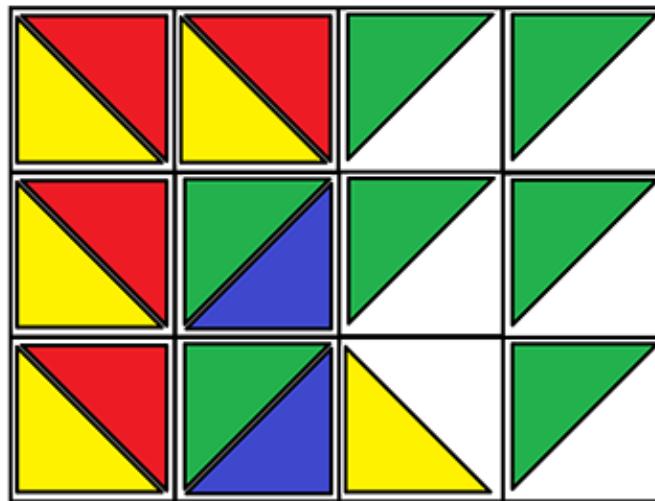


Рис. 2.2.2

***Решение***

Ниже представлено решение на языке C++.

**C++**

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int a, b, c, d, n, m;
6     cin >> a >> b >> c >> d >> n >> m;
7     if(a > c){
8         swap(a, c);
9     }
10    if(b > d){
11        swap(b, d);
12    }
13    int f = a + b;
14    int k = n * m;
15    if(k <= f){
16        cout << k * 2;
17        return 0;
18    }
19    k -= f;
20    c -= a;
21    d -= b;
22    cout << f * 2 + min(k, c + d) << endl;
23 }
```

***Задача 2.2.2.2. Старая задача на новый лад (15 баллов)***

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

### **Условие**

Одна старая задача имеет следующий вид:

«Разбить число 45 на сумму четырех слагаемых так, что если к первому прибавить 2, из второго вычесть 2, третье умножить на 2, а четвертое разделить на 2, то получится одно и то же число».

Ответ к этой задаче — четыре числа 8, 12, 5 и 20. Можно убедиться, что в сумме они дают число 45, а если с каждым из них проделать соответствующую арифметическую операцию, то получится одно и то же число 10.

Необходимо решить чуть более общую задачу: даны числа  $n$  и  $k$ . Нужно представить число  $n$  в виде суммы четырех целых неотрицательных слагаемых  $a + b + c + d$  таких, что  $a + k = b - k = c \cdot k = d / k$ . Гарантируется, что для заданных  $n$  и  $k$  такое разбиение существует.

### **Формат входных данных**

В одной строке через пробел два числа  $n$  и  $k$ , где  $1 \leq n \cdot k \leq 10^{18}$ .

### **Формат выходных данных**

Вывести через пробел в одну строку четыре целых неотрицательных числа  $a, b, c, d$  таких, что  $a + b + c + d = n$  и  $a + k = b - k = c \cdot k = d / k$ .

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
45 2
<b>Стандартный вывод</b>
8 12 5 20

#### *Пример №2*

<b>Стандартный ввод</b>
128 7
<b>Стандартный вывод</b>
7 21 2 98

## **Решение**

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int n, k;
6     cin >> n >> k;
7     int x = (k * n) / (k * k + 2 * k + 1);
8     cout << x - k << ' ' << x + k << ' ' << x / k << ' ' << x * k << endl;
9 }
```

## **Задача 2.2.2.3. Ладья и обязательная клетка (20 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

### **Условие**

Шахматная ладья находится в левом верхнем углу прямоугольного поля, разбитого на клетки размером  $n \times m$ .  $n$  обозначает число строк,  $m$  — число столбцов. Она хочет попасть в правую нижнюю клетку этого поля кратчайшим путем. Ладья может передвигаться либо вправо, либо вниз на любое количество клеток. Ладья обязана посетить заданную клетку с координатами  $(x, y)$ , где  $x$  — номер строки этой клетки, а  $y$  — номер ее столбца.

Требуется найти количество способов построить путь ладьи из левого верхнего угла в правый нижний, которые проходят через обязательную клетку с заданными координатами.

### **Формат входных данных**

В первой строке находятся два числа через пробел:  $n$  — число строк и  $m$  — число столбцов прямоугольного поля,  $2 \leq n, m \leq 25$ . Во второй строке через пробел находятся координаты  $(x, y)$  обязательной для посещения клетки, где  $1 \leq x \leq n$ ,  $1 \leq y \leq m$ . Координаты  $x$  и  $y$  не совпадают с координатами левой верхней и правой нижней клеток.

### **Формат выходных данных**

Вывести одно число — количество кратчайших путей ладьи из верхней левой в правую нижнюю клетку, проходящих через заданную клетку.

## Примеры

Стандартный ввод
3 4
2 3
Стандартный вывод
6

## Примечания

На рис. 2.2.3 представлены шесть путей, которыми ладья может пройти по полю размером  $3 \times 4$ , обязательно посещая по пути клетку (2, 3).

## Комментарий

Задачу можно решить как комбинаторными методами (произведение биномиальных коэффициентов), так и динамическим программированием.

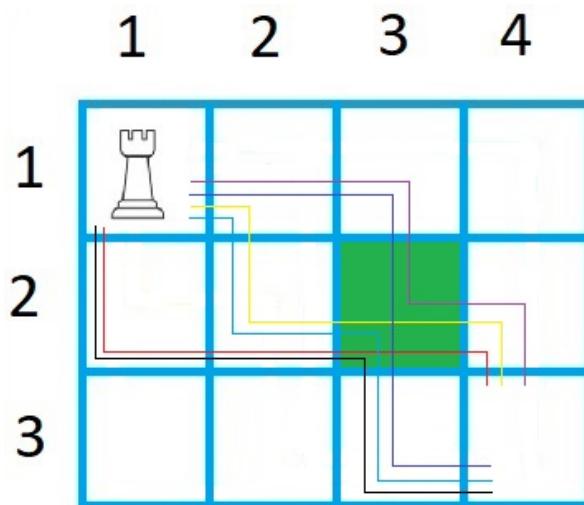


Рис. 2.2.3

## Решение

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     vector<vector<int>> bc(51, vector<int>(51, 0));
6     bc[0][0] = 1;
7     for(int i = 1; i <= 50; i++){
8         for(int j = 0; j < 51; j++){

```

```

9         bc[i][j] += bc[i - 1][j];
10        if(j - 1 >= 0){
11            bc[i][j] += bc[i - 1][j - 1];
12        }
13    }
14
15    int n, m, x, y;
16    cin >> n >> m >> x >> y;
17    int d1 = bc[x - 1 + y - 1][x - 1];
18    int d2 = bc[n - x + m - y][n - x];
19    int ans = d1 * d2;
20    cout << ans << endl;
21 }
```

#### **Задача 2.2.2.4. Танец с цифрами (25 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Десять танцоров репетируют на сцене новый танец. Каждый танцор одет в футбольку, на которой написана одна из цифр от 1 до 9, цифры могут повторяться. Изначально они стоят в некотором порядке слева направо, и их цифры образуют некоторое десятизначное число  $A$ . Далее во время всего танца участники либо разбиваются на пять пар рядом стоящих танцоров и одновременно меняются местами внутри своих пар, либо самый левый танцор перемещается на самую правую позицию и становится самым правым танцором.

Сын постановщика танца от скучки на бумаге выписывает все получающиеся при каждом перемещении десятизначные числа. Так как танец длинный, то в итоге на бумаге окажутся все возможные числа, которые в принципе могут появится при этих условиях. Нужно найти разницу между самым большим и самым маленьким из этих чисел.

#### **Формат входных данных**

На вход подается одно десятизначное число  $A$ , обозначающее начальное расположение танцоров. В числе могут встречаться цифры от 1 до 9, некоторые из них могут повторяться.

#### **Формат выходных данных**

Вывести одно число, равное разности самого большого и самого маленького из чисел, которые могут быть получены во время танца.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
1456531355
<b>Стандартный вывод</b>
5182160085

## Примечания

Самое маленькое число, которое можно получить в примере, равно 1353155456, самое большое равно 6535315541.

Покажем, как получить эти числа из исходного числа 1456531355. Сначала получим самое большое следующим образом: две левые цифры, 1 и 4, переместим вправо, получим 5653135514, потом поменяем в парах цифры местами и получим самое большое — 6535315541. Далее опять поменяем порядок в парах и в числе 5653135514 переместим три левых цифры 5, 6 и 5 вправо, получим 3135514565 и здесь снова поменяем порядок в парах, получим самое маленькое — 1353155456. Таким образом, искомая разница равна 5182160085.

## Решение

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     string s;
6     cin >> s;
7     string mx = s, mn = s;
8
9     for(int i = 0; i < 5; i++){
10         for(int j = 0; j < 10; j++){
11             mx = max(mx, s);
12             mn = min(s, mn);
13             if(j < 9){
14                 s = s.substr(1) + s[0];
15             }
16         }
17         for(int j = 0; j < 5; j++){
18             swap(s[2 * j], s[2 * j + 1]);
19         }
20     }
21     stringstream ssmn;
22     ssmn << mn;
23     int imn;
24     ssmn >> imn;
25     stringstream ssmx;
```

```

26     ssmx << mx;
27     int imx;
28     ssmx >> imx;
29     cout << imx - imn << endl;
30 }
```

### **Задача 2.2.2.5. Трудная сортировка (30 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 3 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Иннокентий работает в отделе сортировки перестановок, подотделе сортировки вставками. Его задача заключается в сортировке перестановок, предоставленных заказчиками. Перестановкой длины  $n$  называется такая последовательность чисел, в которой встречаются все числа от 1 до  $n$  без повторений в некотором порядке.

Перестановка считается отсортированной, если в ней все числа расположены по возрастанию, то есть она имеет вид  $1, \dots, n$ .

Иннокентий начинает рабочий день с пустой последовательности чисел. За день он сортирует вставками перестановку длины  $n$ . В начале каждой операции вставки он получает очередное число  $a_i$  из перестановки заказчика, после чего обрабатывает его, вставляя в отсортированную последовательность из ранее полученных чисел. После каждого такого добавления последовательность уже обработанных чисел должна быть отсортирована по возрастанию.

Перед тем как вставить число  $a_i$  в последовательность, он может выбрать, с какого края последовательности начать вставку. Далее он устанавливает число  $a_i$  с этого края и последовательно меняет вставляемое число с рядом стоящим числом  $b_j$  до тех пор, пока число  $a_i$  не встанет на свое место. На каждую перестановку вставляемого числа  $a_i$  с числом  $b_j$  Иннокентий тратит  $b_j$  единиц энергии.

Дана перестановка длины  $n$  из чисел  $a_i$  в том порядке, в котором Иннокентий их будет обрабатывать. Подскажите ему, какое минимальное количество энергии ему потребуется потратить, чтобы отсортировать всю перестановку.

#### **Формат входных данных**

В первой строке находится одно целое число  $n$  — длина перестановки, где  $1 \leq n \leq 2 \cdot 10^5$ .

Во второй строке содержится  $n$  целых чисел  $a_i$  через пробел в том порядке, в котором они поступают на обработку Иннокентию. Гарантируется, что эти числа образуют перестановку длины  $n$ , то есть каждое число от 1 до  $n$  содержится в заданном наборе ровно один раз.

## **Формат выходных данных**

Вывести одно число — минимальные суммарные энергозатраты Иннокентия для сортировки вставками заданной на входе перестановки.

## **Примеры**

### *Пример №1*

<b>Стандартный ввод</b>
9 2 9 1 5 6 4 3 8 7
<b>Стандартный вывод</b>
43

## **Примечания**

Первым устанавливается число 2. Оно ни с чем не меняется местами, поэтому затрат нет.

Далее устанавливается число 9. Выбираем правый край и ставим его туда без потерь энергии.

Затем устанавливаем число 1. Выбираем левый край, ставим его туда и снова потерь нет.

Теперь нужно вставить число 5. Если его вставлять с правого края, придется менять местами с 9, а если с левого, то с 1 и 2, что суммарно явно лучше. Итого затраты на вставку 5 равны 3.

Число 6 снова лучше вставить слева, затраты на его вставку равны 8.

Число 4 вставим слева за 3.

Число 3 так же слева за 3.

А вот число 8 лучше вставить справа за 9.

И осталось число 7. Если вставлять слева, то затратим 21, а если справа, то всего 17.

Итого на сортировку заданной перестановки потратили:  $0 + 0 + 0 + 3 + 8 + 3 + 3 + 9 + 17 = 43$ .

## **Комментарий**

Построим дерево отрезков на сумму, при обработке числа  $a$  будем находить, какая сумма на данный момент меньше: от 1 до  $a - 1$  или от  $a + 1$  до  $n$ . Прибавим ее к ответу и поместим в позицию  $a$  это число  $a$ .

## Решение

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int LG = 19;
5 int N = (1 << LG);
6 vector<int> tr(2 * N, 0);
7 void upd(int pos, int x){
8     pos += N;
9     tr[pos] = x;
10    pos /= 2;
11    while(pos){
12        tr[pos] = {tr[2 * pos]+ tr[2 * pos + 1]};
13        pos /= 2;
14    }
15 }
16 int get(int l, int r){
17     l += N;
18     r += N;
19     int res = 0;
20     while(l <= r){
21         if(l % 2 == 1){
22             res += tr[l];
23         }
24         if(r % 2 == 0){
25             res += tr[r];
26         }
27         l = (l + 1) / 2;
28         r = (r - 1) / 2;
29     }
30     return res;
31 }
32 signed main(){
33     int n, a;
34     cin >> n;
35     int ans = 0;
36     for(int i = 0; i < n; i++){
37         cin >> a;
38         int sl = get(0, a - 1);
39         int sr = get(a + 1, N - 1);
40         ans += min(sl, sr);
41         upd(a, a);
42     }
43     cout << ans << endl;
44 }
```

## 2.2.3. Третья волна. Задачи 8–11 класса

Задачи третьей волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63456/enter/>.

### **Задача 2.2.3.1. Туннель (10 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### ***Условие***

Рассмотрим классическую задачу прохождения группы с одним фонариком по туннелю. Есть четыре человека, и у них есть один фонарик. Нужно перевести всю группу на другой конец туннеля. По туннелю можно проходить только с фонариком и только либо вдвоем, либо в одиночку. По этой причине придется сделать пять рейсов по туннелю: три рейса туда и два рейса обратно. Туда идут двое, обратно — один, возвращая фонарик еще не прошедшей части группы. У каждого из четырех человек своя скорость передвижения по туннелю, но некоторые скорости могут совпадать. Двое идут со скоростью самого медленного в этой паре. Нужно найти минимальное время, за которое можно перевести группу по туннелю.

Здесь, в зависимости от скоростей персонажей, есть две стратегии. Проиллюстрируем их на примерах.

Пусть есть люди  $A, B, C, D$ . У  $A$  — время прохождения туннеля 1 мин, у  $B$  — 4 мин, у  $C$  — 5 мин, у  $D$  — 10 мин. Здесь работает наиболее очевидная стратегия: самый быстрый переводит текущего и возвращается с фонариком обратно за следующим. При этой стратегии нужно проходить так:

- $A, B$  туда, затрачено 4 мин;
- $A$  обратно, затрачена 1 мин;
- $A, C$  туда, затрачено 5 мин;
- $A$  обратно, затрачена 1 мин;
- $A, D$  туда, затрачено 10 мин.

Общее время  $4 + 1 + 5 + 1 + 10 = 21$  мин.

Но не всегда эта стратегия оптимальна. Уменьшим время прохождения туннеля персонажем  $B$  до 2 мин. По вышеопределенной стратегии будет 19 мин ( $2 + 1 + 5 + 1 + 10 = 19$ ), но имеется более быстрое решение:

- $A, B$  туда, затрачено 2 мин;
- $A$  обратно, затрачена 1 мин;
- $C, D$  туда, затрачено 10 мин;
- $B$  обратно, затрачено 2 мин;
- $A, B$  туда, затрачено 2 мин.

Общее время  $2 + 1 + 10 + 2 + 2 = 17$  мин.

Заметим, что для предыдущего примера такая стратегия не работает:  $4 + 1 + 10 + 4 + 4 = 23$  мин.

Если же персонаж  $B$  проходит туннель за 3 мин (а все остальные так же, как и в примерах), то независимо от стратегии будет затрачено 20 мин. В этом случае считаем, что работает первая стратегия.

Поразмыслив, станет понятно, от какого условия зависит выбор стратегии. Далее будем всегда считать, что  $A$  движется не медленнее  $B$ ,  $B$  движется не медленнее  $C$ ,  $C$  движется не медленнее  $D$ .

Дано время прохождения туннеля персонажами  $A, C, D$ . Нужно найти границу `border` для  $B$  такую, что если определить для  $B$  время прохождения строго меньшее, чем `border`, то выгодна вторая стратегия, иначе — первая.

### **Формат входных данных**

В одной строке задано три целых чисел через пробел — время прохождения туннеля персонажами  $A, C, D$ . Времена даны по неубыванию. Все числа на входе в пределах от 1 до 100.

### **Формат выходных данных**

Вывести одно число — границу `border` для  $B$  такую, что если определить время прохождения им туннеля строго меньше, чем `border`, нужно использовать вторую стратегию, иначе — первую. Ответ может быть нецелым, поэтому вывести его нужно с одним знаком после десятичной точки.

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
1 5 10
<b>Стандартный вывод</b>
3

### **Решение**

Ниже представлено решение на языке C++.

<b>C++</b>
<pre> 1 #include&lt;bits/stdc++.h&gt; 2 #define int long long 3 using namespace std; 4 signed main(){ 5     int A, C, D; 6     cin &gt;&gt; A &gt;&gt; C &gt;&gt; D; 7     cout.precision(1); 8     cout &lt;&lt; fixed &lt;&lt; (A + C) / 2.0 &lt;&lt; endl; 9 }</pre>

### **Задача 2.2.3.2. Математический пазл (15 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### ***Условие***

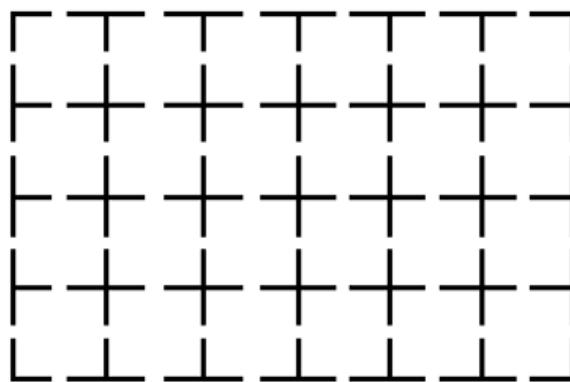


Рис. 2.2.4

Компания по производству пазлов решила освоить принципиально новый тип головоломок. Для этого берется прямоугольная решетка размера  $n \times m$ , каждый ее столбец и строка разрезаются посередине пополам. После этого образуются фигуры трех типов: четыре уголка,  $2 \cdot (n+m-2)$  т-образных фигур и  $(n-1) \cdot (m-1)$  крестиков.

Тому, кто решает головоломку, требуется сложить из этих фигур исходную прямоугольную решетку. При этом необходимо использовать абсолютно все имеющиеся в наличии фигуры.

#### ***Формат входных данных***

В первой строке заданы через пробел два числа  $a$  — количество т-образных фигур и  $b$  — количество крестиков, которые находятся в одном из пазлов. При этом в наборе всегда есть еще четыре уголка. Известно, что этот комплект позволяет собрать прямоугольную решетку размера  $n \times m$ , где  $1 \leq n, m \leq 10^9$ .

#### ***Формат выходных данных***

Требуется по числам  $a$  и  $b$  найти размеры исходной решетки  $n$  и  $m$ . Будем всегда считать, что  $n \leq m$ , то есть нужно вывести в одну строку через пробел два числа, первое из которых не превосходит второго, и вместе они задают размеры загаданной решетки.

## Примеры

*Пример №1*

Стандартный ввод
16 15
Стандартный вывод
4 6

*Пример №2*

Стандартный ввод
0 0
Стандартный вывод
1 1

## Комментарий

Задачу можно решить либо бинарным поиском, либо при помощи квадратного уравнения.

## Решение

Ниже представлено решение на языке C++ при помощи бинпоиска.

C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int a, b;
6     cin >> a >> b;
7     int L = 0, R = a / 4 + 1;
8     while(R - L > 1){
9         int M = (R + L) / 2;
10        int D = a / 2 - M;
11        if(M * D <= b){
12            L = M;
13        }
14        else{
15            R = M;
16        }
17    }
18    cout << L + 1 << ' ' << a / 2 - L + 1 << endl;
19 }
```

### **Задача 2.2.3.3. Восемь пирогов и одна свечка (20 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### ***Условие***

Мечта Карлсона наконец-то сбылась! Мама Малыша испекла восемь пирогов прямоугольной формы и в один из них воткнула свечку. После того как Карлсон съел семь пирогов, он решил-таки поделиться кусочком оставшегося восьмого пирога с Малышом. Но, будучи в хорошем настроении, он вынул из пирога свечу и предложил ему решить задачку.

«Так как я самый щедрый Карлсон в мире, то делить оставшийся пирог будешь ты. Но учи, ты должен разрезать пирог одним прямым разрезом так, чтобы линия прошла через один из углов и точку, где стояла свечка. После этого я выберу себе один из двух кусочков, а оставшийся, так и быть, достанется тебе».

Малыш не против этого замысла, однако считает, что разрезать пирог нужно как можно более справедливо, то есть так, чтобы разница между меньшим и большим кусками была как можно меньше. Подскажите Малышу, какой минимальной разницы между площадями кусков он сможет добиться.

#### ***Формат входных данных***

В первой строке находятся два числа  $n$  и  $m$  через пробел — размеры прямоугольного пирога. Пирог размещен на координатной плоскости так, что его левый нижний угол находится в точке  $(0, 0)$ , а правый верхний — в точке  $(n, m)$ , где  $2 \leq n, m \leq 1000$ .

Во второй строке находятся два числа  $x$  и  $y$  через пробел — координаты свечки, где  $1 \leq x \leq n - 1$ ,  $1 \leq y \leq m - 1$ , то есть свечка находится строго внутри пирога.

#### ***Формат выходных данных***

Вывести одно вещественное число с точностью не менее трех знаков после десятичной точки — минимальную разницу между площадями двух получающихся после разрезания кусков, которую сможет получить Малыш.

## Примеры

### Пример №1

<b>Стандартный ввод</b>
8 5
7 2
<b>Стандартный вывод</b>
12.571

### Пример №2

<b>Стандартный ввод</b>
2 2
1 1
<b>Стандартный вывод</b>
0.000

## Примечания

На рис. 2.2.5 представлены четыре варианта разделения пирога для первого примера из условия. Можно видеть, что самый близкий к справедливому способ разделения связан с разрезом из левого верхнего угла. Площадь треугольника в этом случае будет равна  $96 / 7$ , площадь четырехугольника равна  $184 / 7$ , и разница равна  $88 / 7$ , что при округлении до трех знаков равно 12,571.

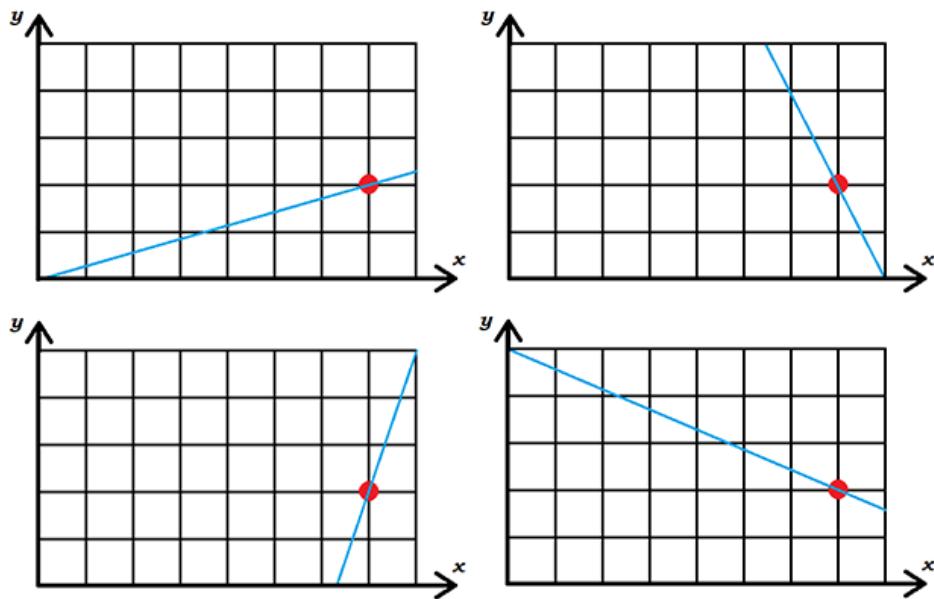


Рис. 2.2.5

## Комментарий

Геометрия: для каждого из четырех случаев аккуратно находим катеты прямоугольного треугольника при помощи пропорции, затем находим площадь этого треугольника и, вычитая из всего прямоугольника эту площадь, находим площадь второго куска. Далее выбираем наиболее оптимальное отношение площадей.

## Решение

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int INF = 1e18;
5 double katy(double x, double y, double n){
6     return n * y / x;
7 }
8 double n, m, x, y;
9 double ans = INF;
10 double k1, k2;
11 void upd(){
12     if(k1 < m){
13         double st = k1 * n / 2;
14         ans = min(ans, n * m - 2 * st);
15     }
16     else{
17         double st = k2 * m / 2;
18         ans = min(ans, n * m - 2 * st);
19     }
20 }
21 signed main(){
22     cin >> n >> m >> x >> y;
23     k1 = katy(x, y, n);
24     k2 = katy(y, x, m);
25     upd();
26     k1 = katy(n - x, y, n);
27     k2 = katy(y, n - x, m);
28     upd();
29     k1 = katy(x, m - y, n);
30     k2 = katy(m - y, x, m);
31     upd();
32     k1 = katy(n - x, m - y, n);
33     k2 = katy(m - y, n - x, m);
34     upd();
35     cout.precision(3);
36     cout << fixed << ans << endl;
}

```

## Задача 2.2.3.4. Плетенка (25 баллов)

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

### **Условие**

У Маши есть  $n$  полосок бумаги.  $i$ -я полоска имеет ширину 1 и длину  $a_i$ . Маша разделит эти полоски на две части и покрасит некоторые в желтый, а оставшиеся — в зеленый цвет. Она сама выберет, какие полоски как покрасить. Далее она хочет из этих полосок сплести максимально большую плетенку. Она расположит полоски одного цвета в некотором порядке горизонтально, а полоски другого цвета в некотором порядке вертикально. После этого она переплетет горизонтальные и вертикальные полоски так, что они будут чередоваться то сверху, то снизу, образуя в местах пересечения шахматную раскраску. Наконец, она обрежет выступающие края полосок так, что останется прямоугольная плетенка с ровными краями. Каждая клетка полученной плетенки должна иметь два слоя.

Маша хочет сплести максимально большую по площади прямоугольную плетенку. Подскажите ей, плетенку какой площади она сможет сделать. Заметим, что она может при создании плетенки использовать не все имеющиеся у нее полоски.

### **Формат входных данных**

В первой строке на вход подается число  $n$  — количество полосок бумаги у Маши, где  $2 \leq n \leq 2 \cdot 10^5$ . Во второй строке через пробел заданы  $n$  целых чисел  $a_i$  через пробел — длины полосок, где  $1 \leq a_i \leq 10^9$ .

### **Формат выходных данных**

Вывести одно число — площадь прямоугольника, форму которого может иметь самая большая плетенка Маши.

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
8
3 6 5 4 4 5 5 2
<b>Стандартный вывод</b>
12

### **Примечания**

На рис. 2.2.6 представлен один из вариантов получения самой большой плетенки для полосок из примера. Синим обозначена граница полученной максимальной плетенки. Ее размер  $3 \times 4$ , и ее площадь 12. При ее создании Маша не должна использовать полоску номер 8, по этой причине неважно, как она раскрашена.

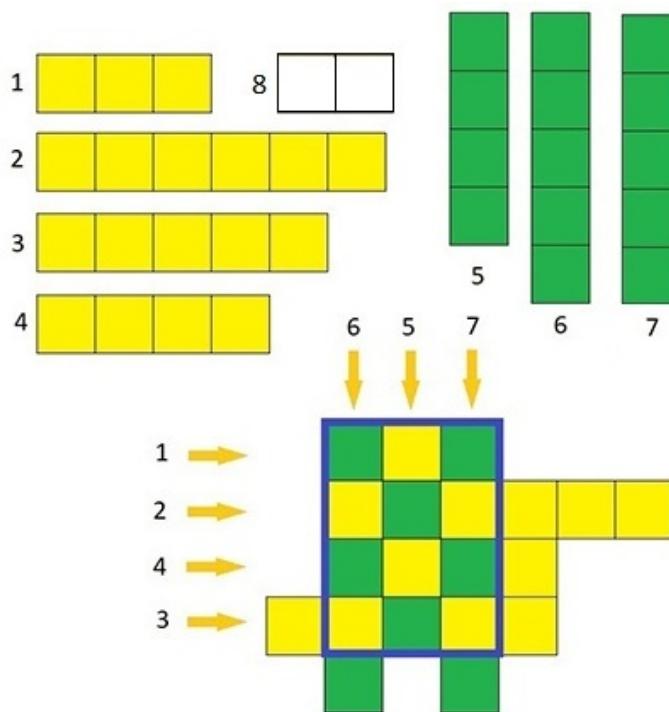


Рис. 2.2.6

***Решение***

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int n;
6     cin >> n;
7     deque<int> v(n);
8     for(int i = 0; i < n; i++){
9         cin >> v[i];
10    }
11    sort(v.begin(), v.end());
12    int ans = 0;
13    int cnth = 0, minh;
14    while(1){
15        if(v.size() == 0){
16            break;
17        }
18        cnth++;
19        minh = v.back();
20        v.pop_back();
21        while(v.size() > 0 && v[0] < cnth){
22            v.pop_front();
23        }
24        ans = max(ans, cnth * min(minh, (int)v.size()));
25    }
26    cout << ans << endl;
27 }
```

### **Задача 2.2.3.5. Английский в игровой форме (30 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 3 с.

**Ограничение по памяти:** 64 Мбайт.

#### ***Условие***

Маша и Витя запоминают слова английского языка в оригинальной игровой форме. За день им нужно выучить  $n$  слов, где  $20 \leq n \leq 100$ , каждое из которых имеет длину от 5 до 8 символов. Маша выбирает из этого набора наугад несколько попарно различных слов (также от 5 до 8) и собирает их в одну строку без пробелов. Далее она переставляет буквы в этой строке так, что слова оказываются полностью перепутанными, и дает эту строку Вите. Теперь Витя должен восстановить все слова, которые выбрала Маша.

Но у Вити плохо получается, а Маша уже забыла, какие слова она выбрала. Нужно им помочь — написать программу, которая восстановит слова, выбранные Машей.

#### ***Формат входных данных***

В первой строке находится строка, которую Маша предложила Вите. Во второй строке содержится число  $n$  — количество слов, которые нужно выучить детям,  $20 \leq n \leq 100$ .

В следующих  $n$  строках содержатся эти слова по одному в строке. Все слова в этом наборе различны. Слова отсортированы в лексикографическом (алфавитном) порядке. Все слова состоят из маленьких букв от *a* до *z*. Обратите внимание, что в тестах к этой задаче все заданные слова реально существуют в английском языке и случайным образом выбраны из словаря.

Гарантируется, что длина каждого слова из предложенного набора (словаря) в пределах от 5 до 8, строка, которую получила Маша, может быть получена путем перестановки букв некоторых различных слов из предложенного словаря, причем, набор выбранных Машей слов определяется по ней однозначно. Количество слов, из которых составлена Машина строка, находится в пределах от 5 до 8.

#### ***Формат выходных данных***

Вывести все слова, выбранные Машей, в алфавитном порядке по одному в строке.

## **Примеры**

### *Пример №1*

#### **Стандартный ввод**

```
stirbaexsudueoeidgomttcrnrwlunapntetacwri
24
bridge
cranky
document
drawing
farmer
fighter
figurine
gravy
havoc
minimum
reactant
reply
republic
sonata
soprano
split
subset
tailor
texture
tomorrow
trout
vicinity
wrist
writer
```

#### **Стандартный вывод**

```
document
drawing
republic
sonata
texture
wrist
```

## **Комментарий**

В случае, выделенном в условии (слова являются случайными, взятыми из английского словаря), задача решается рекурсией с перебором вариантов.

## **Решение**

Ниже представлено решение на языке C++.

## C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 string frs;
5 int n;
6 vector<string> dict;
7 vector<int> msk(26, 0);
8 int cnt = 0;
9 vector<vector<int>> amsk;
10 vector<string> ans;
11 bool bigok = 0;
12 void p(int pos){
13     if(!bigok){
14         if(cnt == 0){
15             sort(ans.begin(), ans.end());
16             bigok = 1;
17             return;
18         }
19         for(int i = pos; i < n; i++){
20             string ts = dict[i];
21             bool ok = 1;
22             for(int j = 0; j < 26; j++){
23                 if(amsk[i][j] > msk[j]){
24                     ok = 0;
25                 }
26             }
27             if(ok){
28                 ans.push_back(ts);
29                 for(int j = 0; j < 26; j++){
30                     msk[j] -= amsk[i][j];
31                     cnt -= amsk[i][j];
32                 }
33                 p(i + 1);
34                 if(!bigok){
35                     for(int j = 0; j < 26; j++){
36                         msk[j] += amsk[i][j];
37                         cnt += amsk[i][j];
38                     }
39                     ans.pop_back();
40                 }
41             }
42         }
43     }
44 }
45 signed main(){
46     cin >> frs;
47     cin >> n;
48     amsk.resize(n, vector<int>(26, 0));
49
50     string ts;
51     for(int i = 0; i < n; i++){
52         cin >> ts;
53         dict.push_back(ts);
54     }
55     for(int i = 0; i < n; i++){
56         for(auto el : dict[i]){
57             amsk[i][el - 'a']++;
58         }
59     }
}

```

```

60     for(auto el : frs){
61         msk[el - 'a']++;
62         cnt++;
63     }
64     p(0);
65     for(auto el : ans){
66         cout << el << endl;
67     }
68 }
```

## 2.2.4. Четвертая волна. Задачи 8–11 класса

Задачи четвертой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63457/enter/>.

### **Задача 2.2.4.1. Квадратный флаг (10 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Одному портному заказали сделать одноцветный флаг. Особенность этого флага в том, что он должен быть квадратным. У портного есть два прямоугольных куска ткани заданного цвета. Один из них имеет размеры  $a \times b$ , другой —  $c \times d$ . Так как клиент будет платить пропорционально площади изготовленного флага, портной хочет сначала сшить имеющиеся у него прямоугольные куски, соединив их двумя какими-то сторонами, а затем из полученного полотна вырезать и сделать флаг с максимально большой стороной. Определить сторону получившегося у него флага.

#### **Формат входных данных**

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника — целые числа  $a, b$  через пробел, во второй — размеры второго прямоугольника, также целые числа  $c, d$  через пробел, где  $1 \leq a, b, c, d \leq 10^9$ .

#### **Формат выходных данных**

Вывести одно число — сторону самого большого квадрата, который можно получить по условию задачи.

## Примеры

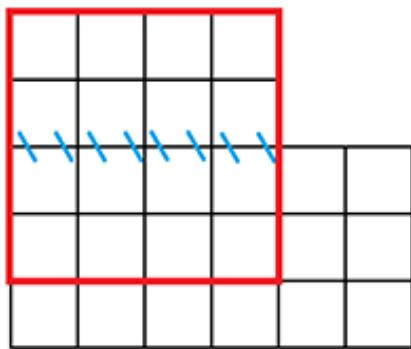
*Пример №1*

Стандартный ввод
2 4 3 6
Стандартный вывод
4

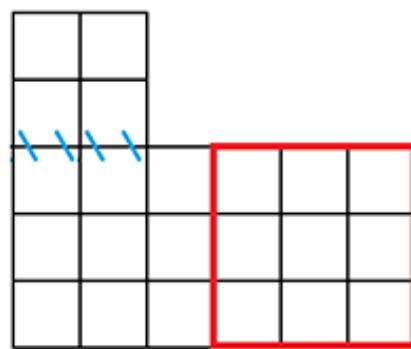
*Пример №2*

Стандартный ввод
2 2 3 6
Стандартный вывод
3

## Примечания



$$2 \times 4 + 3 \times 6$$



$$2 \times 2 + 3 \times 6$$

Рис. 2.2.7

На рис. 2.2.7 представлены иллюстрации для тестов из условия. Синими штрихами обозначено место сшивки двух кусков. Красный квадрат выделяет один из вариантов вырезания максимального квадрата.

## Решение

Ниже представлено решение на языке C++.

## C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int a, b, c, d;
6     cin >> a >> b >> c >> d;
7     int ans = max(min(a, b), min(c, d));
8     int p1 = min(a + c, min(b, d));
9     int p2 = min(a + d, min(b, c));
10    int p3 = min(b + c, min(a, d));
11    int p4 = min(b + d, min(a, c));
12    ans = max({ans, p1, p2, p3, p4});
13    cout << ans << endl;
14 }

```

**Задача 2.2.4.2. Потерянная ДНК (15 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

**Условие**

В данной задаче будем упрощенно считать, что ДНК представляется строкой длины от 10 до 100, состоящей из букв А, С, Г, Т.

Пусть даны две ДНК  $D_1$  и  $D_2$  одной и той же длины  $n$ . Выберем некоторое произвольное число  $i$  от 1 до  $n - 1$  и поменяем местами префиксы (начала) этих ДНК длины  $i$ . Будем говорить, что полученные новые две строки образованы путем скрещивания двух исходных по префикску длины  $i$ .

Например, пусть  $D_1 = \texttt{AACGGTAGGT}$ , а  $D_2 = \texttt{TCCCGGAACA}$ . Выберем  $i = 4$  и поменяем местами префиксы длины 4. Получим две новые ДНК, одна из которых будет иметь вид  $\texttt{AACGGAACA}$ , а вторая —  $\texttt{TCCCGTAGGT}$ . Для наглядности были выделены части первой из них.

Полученные новые ДНК снова могут быть скрещены по любому префикску длины от 1 до  $n - 1$ .

Теперь можно рассмотреть популяцию из нескольких ДНК. Выберем из них две, произведем их скрещивание по префикску какой-либо длины и поместим две новые ДНК в исходную популяцию. В данной задаче будем считать, что количество ДНК не увеличивается, то есть старые две ДНК заменяются на новые две ДНК.

Дана исходная популяция из  $t$  ДНК, каждая имеет одну и ту же длину  $n$ . После некоторого количества парных скрещиваний была получена новая популяция. Но при итоговой обработке данных сведения об одной ДНК из новой популяции были потеряны. Задача состоит в отыскании этой потерянной ДНК по оставшимся  $t - 1$  ДНК из новой популяции.

### **Формат входных данных**

В первой строке через пробел даны два числа  $n$  — длина ДНК и  $m$  — количество ДНК в исходной популяции, где  $10 \leq n \leq 100$ ,  $2 \leq m \leq 100$ .

В следующих  $m$  строках содержится описание исходной популяции ДНК, каждая задается строкой длины  $n$ , состоящей из символов А, С, Г и Т.

Далее следует разделяющая строка, содержащая  $n$  символов «—».

Далее следует еще  $m - 1$  строк, описывающих новую (заключительную) популяцию без одной ДНК.

Гарантируется, что данные верны, то есть  $m - 1$  последняя ДНК является некоторой новой популяцией ровно без одной ДНК, полученной из исходной популяции, заданной в  $m$  первых строках.

### **Формат выходных данных**

Вывести недостающую утерянную ДНК.

### **Примеры**

#### *Пример №1*

##### **Стандартный ввод**

```
10 2
AACGGTAGGT
TCCCGGAACA
-----
TCCCGTAGGT
```

##### **Стандартный вывод**

```
AACGGGAACA
```

#### *Пример №2*

##### **Стандартный ввод**

```
10 4
AACCGGTTAA
ACGTACGTAC
AAACCCGGGT
CATTACTGGA
-----
AAGCGCTTAA
CCACACGTGC
AACTAGGGGT
```

##### **Стандартный вывод**

```
AATTCTTGAA
```

## **Комментарий**

Для каждой позиции нужно найти недостающую букву из первого набора ДНК. Для этого удобнее всего использовать функцию `xor`.

## **Решение**

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int n, m;
6     cin >> n >> m;
7     vector<string> v1(m);
8     for(int i = 0; i < m; i++){
9         cin >> v1[i];
10    }
11    string d;
12    cin >> d;
13    vector<string> v2(m - 1);
14    for(int i = 0; i < m - 1; i++){
15        cin >> v2[i];
16    }
17    for(int j = 0; j < n; j++){
18        int ss = 0;
19        for(int i = 0; i < m; i++){
20            ss ^= (int)(v1[i][j]);
21        }
22        for(int i = 0; i < m - 1; i++){
23            ss ^= (int)(v2[i][j]);
24        }
25        cout << (char)(ss);
26    }
27    cout << endl;
28 }
```

## **Задача 2.2.4.3. Утомленные туристы (20 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

## **Условие**

Рассмотрим следующий вариант известной задачи на перемещение по туннелю группы из четырех человек. В общем виде она выглядит так: четыре туриста хотят пройти по темному туннелю. Имеется один фонарик. По туннелю можно перемещаться либо вдвоем, либо по одному, при этом у тех, кто движется в туннеле,

должен быть фонарик в руках. По этой причине движение должно быть следующим: двое переходят туда, один возвращается обратно и приносит фонарик тем, кто еще не перешел. После этого указанный маневр повторяется снова.

У каждого участника своя скорость движения в туннеле. Пусть участники проходят туннель за  $A, B, C$  и  $D$  мин. Если идут двое, то они движутся со скоростью того, кто идет медленнее. Требуется по заданным временем прохождения туннеля каждого из участников перевести их максимально быстро через туннель.

Немного усложним данную задачу. Введем фактор усталости. А именно, любой участник, пройдя по туннелю, устает и в следующий раз идет уже медленнее. После каждого прохождения туннеля время прохождения любого участника увеличивается на  $E$  мин. Например, если участник до начала движения проходит туннель за 1 мин, а показатель усталости  $E$  равен 3 мин, то первый раз участник пройдет туннель за 1 мин, второй раз — за 4 мин, третий раз — за 7 мин и т. д.

По заданным  $A, B, C, D$  и  $E$  узнать, за какое минимальное время можно провести всю группу через туннель согласно указанным правилам.

### **Формат входных данных**

На вход подаются пять чисел. В первой строке через пробел четыре числа  $A, B, C$  и  $D$  — время прохождения туннеля каждым из четырех участников до того, как они начали движение. Во второй строке содержится число  $E$  — величина, на которую увеличивается время прохождения туннеля каждым участником после каждого перемещения. При этом  $1 \leq A, B, C, D \leq 1000$ ,  $0 \leq E \leq 1000$ .

### **Формат выходных данных**

Вывести одно число — минимальное время прохождения туннеля всей группой.

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
8 9 10 1
3
<b>Стандартный вывод</b>
44

#### *Пример №2*

<b>Стандартный ввод</b>
8 9 10 1
0
<b>Стандартный вывод</b>
29

## Примечания

В первом примере при прохождении туннеля каждый турист устает и движется медленнее на 3 мин. Покажем, как перевести группу при этом за 44 мин.

Каждую ситуацию будем обозначать следующим образом: слева от двоеточия находятся туристы, которые стоят в начале туннеля, а справа — те, что стоят в конце туннеля. Туриста будем обозначать при помощи числа, соответствующего его текущему времени прохождения туннеля.

Тогда исходная ситуация имеет вид 1, 8, 9, 10 :.

Сначала идут туристы 1 и 8, каждый после перехода устает на 3 мин, получим ситуацию 9, 10 : 4, 11, затрачено 8 мин.

Обратно возвращается турист 4, он устает еще на 3 мин. Ситуация становится 7, 9, 10 : 11, затрачено  $8 + 4 = 12$  мин.

Теперь идут туристы 7 и 9, получится ситуация 10 : 10, 11, 12, затрачено  $8 + 4 + 9 = 21$  мин.

Возвращается турист 10, получится 10, 13 : 11, 12, затрачено  $8 + 4 + 9 + 10 = 31$  мин.

Наконец, оставшиеся двое туристов 10 и 13 за 13 мин переходят туннель, итого затрачено  $8 + 4 + 9 + 10 + 13 = 44$  мин.

## Комментарий

Задача решается рекурсивным перебором всех вариантов прохождения.

## Решение

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int INF = 1e18;
5 vector<int> v(4);
6 int e, ans = INF;
7 void p(vector<int> &vl, vector<int> &vr, int tv){
8     if(vl.size() == 2){
9         ans = min(ans, tv + *max_element(vl.begin(), vl.end()));
10        return;
11    }
12    for(int i = 0; i < vl.size() - 1; i++){
13        for(int j = i + 1; j < vl.size(); j++){
14            vector<int> vl1;
15            for(int k = 0; k < vl.size(); k++){
16                if(k != i && k != j){
17                    vl1.push_back(vl[k]);
18                }
19            }
20            vector<int> vr1 = vr;

```

```

21         vr1.push_back(vl[i] + e);
22         vr1.push_back(vl[j] + e);
23         int tmp = max(vl[i], vl[j]);
24         sort(vr1.rbegin(), vr1.rend());
25         vl1.push_back(vr1.back() + e);
26         vr1.pop_back();
27         p(vl1, vr1, tv + tmp + vl1.back() - e);
28     }
29 }
30
31 signed main(){
32     for(int i = 0; i < 4; i++){
33         cin >> v[i];
34     }
35     sort(v.begin(), v.end());
36     cin >> e;
37     vector<int> vl = v, vr;
38     p(vl, vr, 0);
39     cout << ans;
40 }
```

### **Задача 2.2.4.4. Проектируем мост (25 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

При постройке моста используются два типа пролетов: П-образные (они прочные, но дорогие) и Т-образные (они дешевле, но менее надежные). Мост должен начинаться и заканчиваться П-образными пролетами. Любой Т-образный пролет должен иметь хотя бы один П-образный пролет в качестве соседнего.

Длина проектируемого моста —  $n$  пролетов. Муниципалитет выделил средства на постройку  $a$  П-образных и  $b$  Т-образных пролетов. При этом  $a + b = n$ . Требуется выяснить, сколькими способами при этих условиях можно скомпоновать мост. Два способа компоновки моста отличаются, если в одной на некоторой позиции стоит П-образный пролет, а в другой на этой же позиции стоит Т-образный пролет.

#### **Формат входных данных**

В одной строке через пробел заданы два числа:  $a$  — число П-образных пролетов и  $b$  — число Т-образных пролетов, на постройку которых выделены средства, где  $2 \leq a \leq 10^6$ ,  $0 \leq b \leq 10^6$ .

## **Формат выходных данных**

Вывести одно число — количество вариантов компоновки моста. Так как ответ может быть очень большим, требуется вывести остаток от его деления на  $1\ 000\ 000\ 007$  ( $10^9 + 7$ ).

## **Примеры**

### *Пример №1*

Стандартный ввод
4 3
Стандартный вывод
7

## **Примечания**

Для примера из условия имеется 7 вариантов компоновки моста (пробелы добавлены для лучшего восприятия вариантов):

П Т Т П Т П П  
 П Т Т П П Т П  
 П Т П Т Т П П  
 П Т П П Т Т П  
 П П Т П Т Т П  
 П П Т Т П Т П  
 П Т П Т П Т П

## **Комментарий**

При заданных ограничениях задача решается только при помощи комбинаторики с вычислениями по модулю.

## **Решение**

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int INF = 1e18;
5 const int MOD = 1e9 + 7;
6 vector<int> f(2e6 + 1, 1);

```

```

7  int binpow (int a, int n) {
8      int res = 1;
9      while (n > 0) {
10         if (n % 2 == 1)
11             (res *= a) %= MOD;
12         (a *= a) %= MOD;
13         n /= 2;
14     }
15     return res;
16 }
17
18 int bc(int n, int k){
19     int res = f[n];
20     int p1 = binpow(f[k], MOD - 2);
21     int p2 = binpow(f[n - k], MOD - 2);
22     (res *= p1) %= MOD;
23     (res *= p2) %= MOD;
24     return res;
25 }
26 signed main(){
27     for(int i = 1; i <= 2e6; i++){
28         f[i] = (f[i - 1] * i) % MOD;
29     }
30     int a, b;
31     int ans = 0;
32     cin >> a >> b;
33     a--;
34     for(int i = 0; i < a + 1; i++){
35         if(2 * i <= b){
36             int d = bc(a, i);
37             if(b - 2 * i <= a - i){
38                 (d *= bc(a - i, b - 2 * i)) %= MOD;
39                 (ans += d) %= MOD;
40             }
41         }
42     }
43     cout << ans << endl;
44 }
```

### **Задача 2.2.4.5. Джентльмены на прогулке (30 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 8 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

По прямому участку улицы, которую будем считать отрезком  $AB$  длины  $d$ , прогуливаются  $n$  джентльменов.  $i$ -й джентльмен движется со скоростью  $v_i$ . Скорости всех джентльменов попарно различны. Дойдя до любого конца улицы, каждый джентльмен поворачивает и идет в обратную сторону.

При каждой встрече два джентльмена приветствуют друг друга, приподнимая головной убор. Приветствие происходит и в том случае, когда один джентльмен обгоняет другого. Если два джентльмена встречаются в момент их одновременного поворота, то происходит два приветствия: одно до поворота, другое — после поворота. Если происходит одновременная встреча трех и более джентльменов, то они приветствуют друг друга попарно, то есть каждый каждого. Допустим, если одновременно встретились четыре джентльмена где-то посреди улицы, произойдет шесть попарных приветствий. Если же эти четыре джентльмена встретились в момент их одновременного поворота, произойдет уже двенадцать приветствий.

В этой задаче считаем, что все действия происходят без остановок, то есть и повороты и приветствия происходят мгновенно. Джентльмены одновременно начинают свою прогулку из точки  $A$  в момент 0. В этот момент они уже производят свои первые попарные приветствия, то есть в момент 0 уже произведено  $n \cdot (n - 1)/2$  приветствий. Момент старта не считается моментом поворота, то есть на старте число приветствий не удваивается. Джентльмены гуляют достаточно долго, чтобы произошло любое заданное количество приветствий.

Требуется найти момент, в который было произведено  $k$ -е по порядку приветствие.

### **Формат входных данных**

В первой строке ввода через пробел содержится два целых числа:  $d$  — длина отрезка  $AB$  и  $n$  — количество прогуливающихся джентльменов, где  $1 \leq d \leq 200$ ,  $2 \leq n \leq 2000$ .

Во второй строке находятся  $n$  целых чисел  $v_i$  через пробел — скорости каждого джентльмена, где  $1 \leq v_i \leq 2000$ . Гарантируется, что все скорости попарно различны. Скорости даны в порядке возрастания, то есть  $v_1 < v_2 < \dots < v_n$ .

В третьей строке содержится одно целое число  $k$  — номер требуемого приветствия, для которого нужно найти момент, когда оно произойдет, где  $1 \leq k \leq 10^9$ .

### **Формат выходных данных**

Вывести одно вещественное число — время, когда произойдет  $k$ -е по порядку приветствие. Ответ вывести с точностью не менее двух знаков после десятичной точки.

### **Примеры**

#### *Пример №1*

Стандартный ввод
5 4
2 5 8 10
6
Стандартный вывод
0.000

*Пример №2*

<b>Стандартный ввод</b>
5 4
2 5 8 10
7
<b>Стандартный вывод</b>
0.556

*Пример №3*

<b>Стандартный ввод</b>
5 4
2 5 8 10
11
<b>Стандартный вывод</b>
1.000

*Пример №4*

<b>Стандартный ввод</b>
5 4
2 5 8 10
15
<b>Стандартный вывод</b>
1.429

*Пример №5*

<b>Стандартный ввод</b>
5 4
2 5 8 10
17
<b>Стандартный вывод</b>
1.667

*Пример №6*

<b>Стандартный ввод</b>
5 4
2 5 8 10
19
<b>Стандартный вывод</b>
1.667

*Пример №7*

Стандартный ввод
5 4 2 5 8 10 21
Стандартный вывод
2.000

*Примечания*

На рис. 2.2.8 приведено положение джентльменов из примеров в моменты времени 0, 1 и 2. Джентльмены обозначены своими скоростями. Стрелками обозначены направления их движения в соответствующий момент. Перечислим и пронумеруем в порядке возрастания моменты попарных приветствий этих джентльменов до момента времени 2 включительно. Если два и более приветствия происходят одновременно, неважно какое из них конкретно имеет номер  $k$ , главное, что они происходят в один и тот же определенный момент времени.

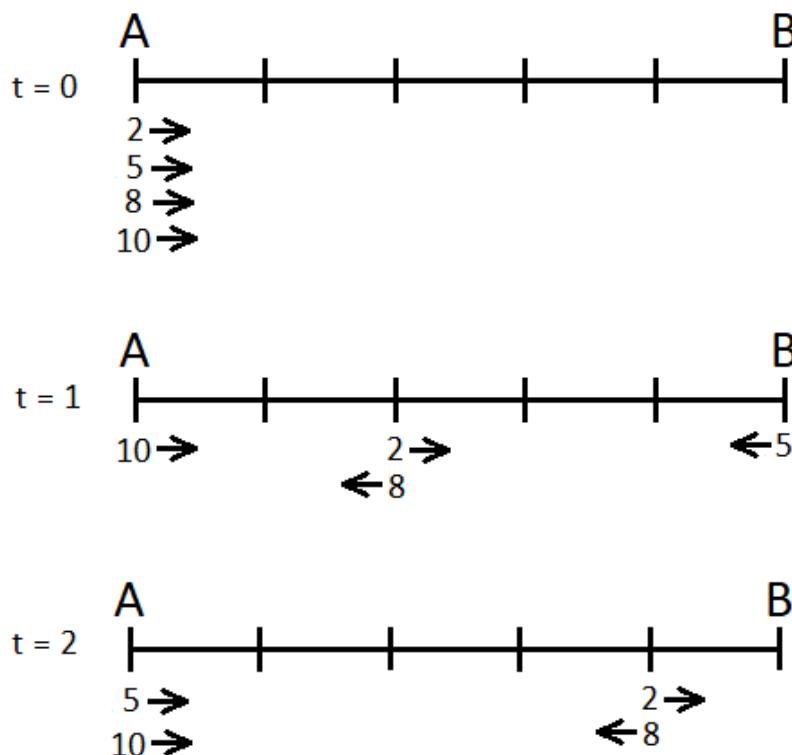


Рис. 2.2.8

1. 2 и 5 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
2. 2 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
3. 2 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
4. 5 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
5. 5 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).

6. 8 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
7. 8 и 10 приветствуют друг друга в момент 0.556.
8. 5 и 10 приветствуют друг друга в момент 0.667.
9. 5 и 8 приветствуют друг друга в момент 0.769.
10. 2 и 10 приветствуют друг друга в момент 0.833.
11. 2 и 8 приветствуют друг друга в момент 1.000 (изображено на рис. 2.2.8).
12. 8 и 10 приветствуют друг друга в момент 1.111.
13. 2 и 10 приветствуют друг друга в момент 1.250.
14. 5 и 10 приветствуют друг друга в момент 1.333.
15. 2 и 5 приветствуют друг друга в момент 1.429.
16. 5 и 8 приветствуют друг друга в момент 1.538.
17. 2 и 8 приветствуют друг друга в момент 1.667.
18. 2 и 10 приветствуют друг друга в момент 1.667.
19. 8 и 10 приветствуют друг друга в момент 1.667 (в момент 1.667 встречаются одновременно три джентльмена 2, 8 и 10).
20. 2 и 8 приветствуют друг друга в момент 2.000 (изображено на рис. 2.2.8).
21. 5 и 10 приветствуют друг друга в момент 2.000 (до поворота).
22. 5 и 10 приветствуют друг друга в момент 2.000 (после поворота, изображено на рис. 2.2.8).

## **Комментарий**

Задача решается при помощи бинпоиска с квадратичным нахождением ответа в каждой его итерации.

## **Решение**

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const double EPS = 1e-7;
5 double x(double M, int V, int d){
6     double dst = V * M;
7     int cnt = floor((dst + EPS) / d);
8     double pin = dst - cnt * d;
9     if(cnt % 2 == 0){
10         return pin;
11     }
12     else{
13         return d - pin;
14     }
15 }
16 int F(double M, vector<int> &v, int d){
17     int res = 0;
18     for(int i = 0; i < v.size(); i++){
19         double dst = v[i] * M;

```

```

20     int cnt = floor((dst + EPS) / d);
21     res += cnt * i;
22     double tx = x(M, v[i], d);
23     for(int j = 0; j < i; j++){
24         double txj = x(M, v[j], d);
25         if(cnt % 2 == 0){
26             res += txj <= tx + EPS;
27         }
28         else{
29             res += txj >= tx - EPS;
30         }
31     }
32 }
33 return res;
34 }
35 signed main(){
36     int d, n;
37     cin >> d >> n;
38     vector<int> v(n);
39     for(int i = 0; i < n; i++){
40         cin >> v[i];
41     }
42     int k;
43     cin >> k;
44     double L = 0, R = 1;
45     while(F(R, v, d) <= k){
46         R *= 2;
47     }
48     R *= 2;
49     while(R - L > 1e-4){
50         double M = (R + L) / 2.0;
51         if(F(M, v, d) < k){
52             L = M;
53         }
54         else{
55             R = M;
56         }
57     }
58     cout.precision(10);
59     cout << fixed << L << endl;
60 }
```

## 2.3. Предметный тур. Математика

### 2.3.1. Первая волна. Задачи 8–9 класса

Задачи первой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Конкурс: <https://contest.yandex.ru/contest/63459/enter/>.

#### **Задача 2.3.1.1. (15 баллов)**

*Тема: арифметика.*

##### **Условие**

Оля в каждую клетку таблицы  $3 \times 3$  записала по некоторому числу и с удивлением заметила, что сумма чисел в каждой строке и в каждом столбце таблицы равна 23. Внимательный же одноклассник Витя к ее размышлениям добавил информацию, что сумма чисел в каждом получившемся квадрате  $2 \times 2$  равна 32. Какое число Оля записала в центральную клетку таблицы?

##### **Решение**

Проанализируем исходную таблицу и увидим, что при построении всех возможных квадратов  $2 \times 2$ :

- числа, стоящие в угловых клетках исходной таблицы, входят по одному разу;
- числа, стоящие во второй строке и во втором столбце — по два раза;
- центральное число — четыре раза.

Тогда если найдем сумму чисел во всех квадратах  $2 \times 2$  и из нее вычтем сумму чисел всей таблицы, а также сумму чисел, стоящих во втором столбце и второй строке, то найдем центральное число, то есть  $32 \cdot 4 - 23 \cdot 3 - 23 \cdot 2 = 13$ .

**Ответ:** 13.

#### **Задача 2.3.1.2. (15 баллов)**

*Тема: комбинаторика.*

##### **Условие**

Нечетное восьмизначное число назовем «интересным», если оно состоит из простых цифр и одинаковые цифры не стоят рядом. Сколько существует таких «интересных чисел»?

***Решение***

Простые цифры — это 2, 3, 5 и 7. Тогда так как «интересное» число должно быть нечетным, то в разряде единиц может стоять только 3, 5 или 7, то есть три варианта. В разряде десятков также может стоять только три варианта, т. к. одинаковые цифры не могут стоять рядом, и т. д. Таким образом, общее количество «интересных» чисел равно  $3^8 = 6561$ .

**Ответ:** 6561.

***Задача 2.3.1.3. (20 баллов)***

*Тема: планиметрия.*

***Условие***

В остроугольном треугольнике  $ABC$  провели высоты  $AA_1$  и  $CC_1$ . Точки  $E$  и  $F$  — середины отрезков  $AC$  и  $A_1C_1$  соответственно.

Найдите длину отрезка  $EF$ , если известно, что  $AC = 30$  и  $A_1C_1 = 24$ .

***Решение***

В прямоугольном треугольнике  $AC_1C$  с гипотенузой  $AC$ :  $C_1E = \frac{1}{2}AC = 15$ .  
Аналогично в треугольнике  $A_1C$ :  $A_1E = \frac{1}{2}AC = 15$ .

Таким образом, треугольник  $A_1C_1E$  является равнобедренным, и его медиана  $EF$  является также и высотой.

Тогда по теореме Пифагора:  $EF^2 = A_1E^2 - A_1F^2 = 15^2 - 12^2 = 81$ ,  $EF = 9$ .

**Ответ:** 9.

***Задача 2.3.1.4. (25 баллов)***

*Темы: уравнения, формулы сокращенного умножения.*

***Условие***

Найдите значение выражения  $x + y + 3z$ , если известно, что числа  $x$ ,  $y$ ,  $z$  удовлетворяют равенству:

$$5x^2 + 4y^2 + 9z^2 + 12z + 13 = 4xy + 12x.$$

***Решение***

Преобразуем равенство следующим образом:

$$(x^2 - 4xy + 4y^2) + (4x^2 - 12x + 9) + (9z^2 + 12z + 4) = 0,$$

то есть

$$(x - 2y)^2 + (2x - 3)^2 + (3z + 2)^2 = 0.$$

Данное равенство будет выполняться при условии, что каждое слагаемое равно 0.

Отсюда получаем систему

$$\begin{cases} x - 2y = 0, \\ 2x - 3 = 0, \\ 3z + 2 = 0, \end{cases}$$

единственным решением которой будет

$$x = \frac{3}{2}; \quad y = \frac{3}{4}; \quad z = -\frac{2}{3}.$$

Тогда

$$x + y + 3z = \frac{3}{2} + \frac{3}{4} + 3 \cdot \left(-\frac{2}{3}\right) = \frac{1}{4} = 0,25.$$

**Ответ:** 0,25.

### Задача 2.3.1.5. (25 баллов)

*Тема: теория вероятностей.*

#### Условие

Шестизначное число будем называть «замечательным», если оно составлено из цифр 1, 2, 3, 4, 5, 6 (каждая цифра используется в числе по одному разу) и кратно 12. Какая вероятность, что сгенерированное компьютером шестизначное число будет «замечательным»?

Ответ выразите в долях и округлите его до четвертого знака после запятой.

#### Решение

Для того чтобы «замечательное» число делилось на 12, оно должно делиться на три и на четыре. Заметим, что все рассматриваемые числа кратны трем, так как сумма их цифр равна 21.

Для того же чтобы число было кратно четырем, необходимо, чтобы две его последние цифры образовывали число, кратное четырем. В нашем случае это могут быть варианты: 12, 16, 24, 32, 36, 52, 56, 64, всего их восемь. К каждому из них нужно приписать впереди четырехзначное число, составленное из остальных четырех цифр, таких чисел  $4! = 24$ . Значит, всего «интересных» чисел  $24 \cdot 8 = 192$ .

Всего же шестизначных чисел  $9 \cdot 10^5 = 900\,000$ .

Тогда вероятность, что сгенерированное компьютером число будет являться «замечательным», будет равна  $\frac{192}{900\,000} \approx 0,0002$ .

**Ответ:** 0,0002.

## 2.3.2. Первая волна. Задачи 10–11 класса

Задачи первой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63476/enter/>.

### **Задача 2.3.2.1. (10 баллов)**

*Темы: комбинаторика, десятичная запись числа, цифры.*

#### ***Условие***

Двухзначное число назовем подходящим, если оно состоит из четных цифр, расположенных по возрастанию (например, 26). Сколько существует таких подходящих чисел?

#### ***Решение***

Число не может начинаться с нуля, так что можно использовать только цифры 2, 4, 6, 8. Выпишем все подходящие: 24, 26, 28, 46, 48, 68.

**Ответ:** 6.

### **Задача 2.3.2.2. (15 баллов)**

*Темы: текстовые задачи, пропорции, составление уравнений.*

#### ***Условие***

На Марсе планируется разместить колонию в 100 тысяч человек. Разные колонисты будут заняты на разных работах и важно, чтобы каждый вид работы выполняли группы из минимального количества человек. Одна из важных задач — обеспечение колонистов сбалансированным питанием. Нормы здорового рациона были рассчитаны таким образом, чтобы обеспечить для каждого человека 350 г картофеля в день. Полный цикл производства картофеля от посадки и до сбора составляет 60 дней, каждые 60 дней часть собранного урожая используется для выращивания нового. В той технологии, которую используют космонавты, с 1 га можно вырастить 250 т картофеля, а для посадки нужно 5 т/га. Специальная обработка почвы позволяет добиться сохранения постоянного уровня урожайности, причем можно засадить и обрабатывать произвольную долю гектара. Чтобы полностью обслуживать один гектар в условиях теплиц на Марсе, требуется труд четырех человек.

Какое минимальное количество человек должны трудиться на выращивании картофеля?

**Решение**

Один человек за 60 дней по плану должен съедать  $60 \cdot 0,35 = 21$  кг картофеля. Следовательно, 100 тысяч человек по плану за это время съедят 2 100 000 кг.

С одного гектара получаем 250 т, но при этом из них 2 т нужно использовать для посадки. Это значит, что с каждого гектара люди получат в свой рацион 245 т картофеля. Если разделить количество картофеля, которое съест по плану колония за 60 дней, на количество картофеля, которое попадет к ним с 1 га, то получится, что требуется приблизительно 8,571 га. Так как каждый гектар должны обрабатывать четыре человека, то для обработки 8,571 га потребуется труд 34,286 человек. Это значит, что 34 человека недостаточно, требуется запланировать труд 35 человек.

**Ответ:** 35.

**Задача 2.3.2.3. (20 баллов)**

*Темы: уравнение параболы, координаты вершины параболы.*

**Условие**

Две параболы с различными вершинами пересекаются таким образом, что первая парабола проходит через вершину второй параболы, а вторая — проходит через вершину первой. Уравнение первой параболы имеет вид  $y = x^2$ , второй  $y = (a \cdot x)^2 + b \cdot x + c$ . Найдите, чему равна величина  $10 \cdot a + c$ .

**Решение**

Координаты вершины первой параболы имеют вид  $(0; 0)$ , следовательно, коэффициент  $c = 0$ . Координаты вершины второй параболы имеют вид

$$\begin{aligned}x &= -\frac{b}{2a}; \\y &= -\frac{b^2}{4a}.\end{aligned}$$

Тогда, подставив их в уравнение первой параболы, получаем:

$$-\frac{b^2}{4a} = \frac{b^2}{2a}.$$

Отсюда  $a = -1$ .

**Ответ:**  $-10$ .

**Задача 2.3.2.4. (25 баллов)**

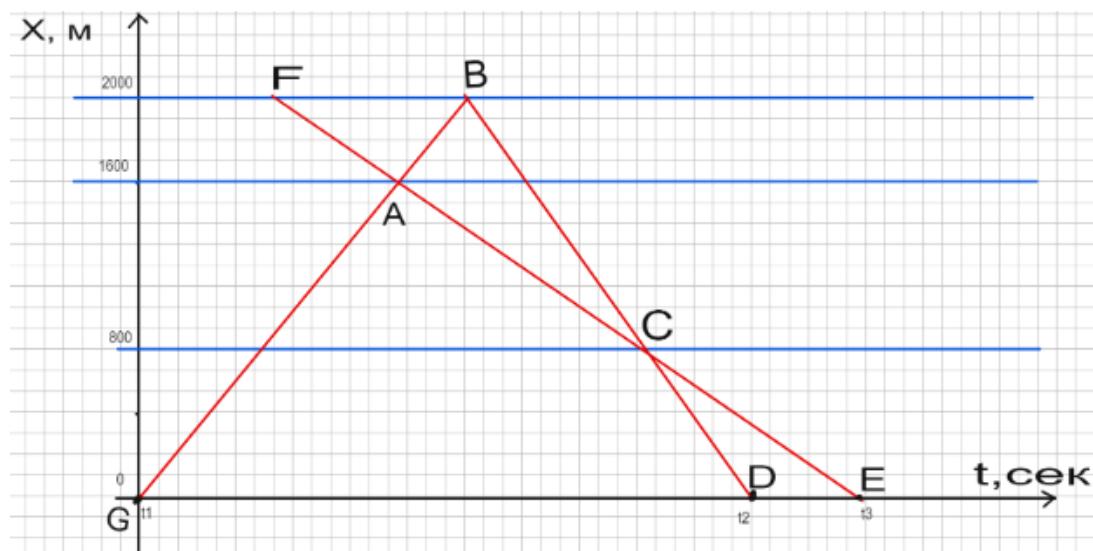
*Темы: текстовые задачи и логика, графическое изображение движения, теорема Менелая.*

### Условие

В 6:00 со дна океана, находящегося на глубине 2 000 м, на поверхность, двигаясь с постоянной скоростью вертикально вверх, начала всплывать подводная лодка. Когда она поднялась до глубины 400 м, капитан заметил, что мимо них вниз плывет глубоководный батискаф. Ему что-то показалось странным. Когда подводная лодка поднялась на поверхность, капитан понял, что на оболочке батискафа были признаки повреждения. Чтобы предотвратить возможную трагедию, в тот же самый момент с подводной лодки вниз спустили спасательный глубоководный аппарат, который спускался с некоторой постоянной скоростью. Когда до дна оставалось 800 м, этот аппарат поравнялся с батискафом. Если бы спасательный аппарат не перехватил батискаф, то спасательный аппарат достиг бы дна к 11:00. Предполагая, что спасательный аппарат все время движения двигался равномерно, определите, в какой момент времени батискаф достиг бы дна, если бы он продолжил движение с той же постоянной скоростью. Ответ введите в виде двух целых чисел, записанных подряд — количество часов и количество минут.

### Решение

Самое изящное решение получается графическим методом.



Здесь данные из условия задачи можно обозначить следующим образом:

$$h_1 = 2000 - 400 = 1600, \quad h_2 = 800, \quad H = 2000, \quad t_1 = 6, \quad t_2 = 12.$$

По теореме Менелая получаем:

$$\frac{GA}{AB} \cdot \frac{BC}{CD} \cdot \frac{DE}{GE} = 1.$$

Выразим эти отношения из разных пар подобных треугольников:

$$\begin{aligned}\frac{GA}{AB} &= \frac{h_1}{H - h_1}; \\ \frac{BC}{CD} &= \frac{H - h_2}{h_2}; \\ \frac{DE}{GE} &= \frac{t_3 - t_2}{t_3 - t_1}; \\ \frac{h_1}{H - h_1} \cdot \frac{H - h_2}{h_2} \cdot \frac{t_3 - t_2}{t_3 - t_1} &= 1.\end{aligned}$$

Подставим числа:

$$\begin{aligned}\frac{1600}{400} \cdot \frac{1200}{800} \cdot \frac{t_3 - 11}{t_3 - 6} &= 1; \\ 6 \cdot (t_3 - 11) &= t_3 - 6; \\ 5 \cdot t_3 &= 60; \\ t_3 &= 12 \text{ ч.}\end{aligned}$$

**Ответ:** 12.

### Задача 2.3.2.5. (30 баллов)

#### Условие

Инженер-исследователь работает над созданием новой системы гиперпространственной навигации для космических кораблей, которая потребует меньших вычислительных ресурсов. Часть измерений гиперпространства скрыта от нас и устроена не так, как мы привыкли, а именно — являются дискретными (с конечным количеством позиций), позиций в которых следуют друг за другом циклически. Например, если это измерение, в котором 5 позиций, то их можно занумеровать числами от 0 до 4 так, что космический корабль, при прямолинейном движении вдоль этого измерения, будет пролетать позиции 0 — 1 — 2 — 3 — 4 — 0 — ... (конечно, корабль в любой момент может изменить направление своего движения на обратное или начать/продолжить изменять позиции и по другим измерениям гиперпространства).

Оказалось, что в гиперпространстве возможна быстрая (но не мгновенная) телепортация: для такого перемещения требуется особая последовательность перемещений в дискретных подпространствах с остановками лишь в выделенные моменты времени. Ранее для хранения таких сложных гипермаршрутов использовалась технология сплошного хранения всех промежуточных опорных точек пути. Однако из-за воздействия агрессивной космической радиации устройства хранения информации часто выходят из строя, что делает сплошное хранение информации очень дорогим, так как требует многократного резервного копирования,

Инженер корабля предложил хранить не сами последовательности позиций, а формулы для их вычисления (что хранить гораздо дешевле и надежнее). В частности, ему удалось запрограммировать движение в одном из измерений с 13 позициями следующим образом: начальное положение обозначается числом 0 и дальнейшие позиции для остановки вычисляются по формуле:  $x_{n+1}$  равно остатку от деления  $(x_n^5 + 2)$  на 13.

Переход корабля из одной позиции в соседнюю по прямому или обратному ходу занимает 1 единицу времени, которую называют таймом. Корабль, используя эту формулу, прошел полный цикл по остановкам и вернулся в позицию с номером 0.

Какое минимальное количество таймов могло занимать все его движение между остановками в ходе этого цикла?

### ***Решение***

Запишем последовательность позиций, в которых останавливается корабль:

$$0 - 2 - 8 - 10 - 6 - 4 - 12 - 1 - 3 - 11 - 9 - 5 - 7 - 0.$$

Между каждыми двумя позициями корабль может двигаться либо прямым ходом, либо обратным. Нужно выбирать кратчайший из двух.

Тогда общая длительность промежутков будет:

$$T = 2 + 6 + 2 + 4 + 2 + 5 + 2 + 2 + 5 + 2 + 4 + 2 + 6 = 44.$$

**Ответ:** 44.

### **2.3.3. Вторая волна. Задачи 8–9 класса**

Задачи второй волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63460/enter/>.

#### ***Задача 2.3.3.1. (15 баллов)***

*Тема: арифметика.*

### ***Условие***

Первый поезд мимо телеграфного столба проезжает за 9 с, второй поезд мимо этого же столба — за 14 с, а, двигаясь навстречу мимо друг друга, они проезжают за 10 с (с момента, когда поравнялись их начала, и до момента, когда разминулись концы).

Во сколько раз скорость первого поезда больше скорости второго?

### ***Решение***

Пусть  $x$  м/с — скорость первого поезда, тогда из условия задачи его длина 9 м. Аналогично, если  $y$  м/с — скорость второго поезда, то его длина равна  $14y$  м.

Зная, что, двигаясь навстречу мимо друг друга, они проезжают за 10 с, составим уравнение:

$$\frac{9x + 14y}{x + y} = 10.$$

Решив это уравнение, получим  $x = 4y$ . То есть скорость первого поезда в четыре раза больше скорости второго.

**Ответ:** 4.

### **Задача 2.3.3.2. (15 баллов)**

*Тема: комбинаторика.*

#### **Условие**

Вася и Петя играют в разведчиков и для этого придумали свой язык шифрования, в котором используются только пять символов. При этом все «слова» в их сообщениях непустые, то есть содержат хотя бы один знак, и длиной не более пяти знаков.

Сколько различных «слов» они имеют в своем арсенале, чтобы передавать друг другу информацию?

#### **Решение**

«Слова», которые могут составлять Вася и Петя на своем языке, могут состоять из 1, 2, 3, 4 и 5 символов.

Тогда общее количество слов будет равно  $5^1 + 5^2 + 5^3 + 5^4 + 5^5 = 3905$ .

**Ответ:** 3905.

### **Задача 2.3.3.3. (20 баллов)**

*Тема: геометрия.*

#### **Условие**

В треугольнике  $ABC$  длина биссектрисы  $AD$  равна длине отрезка  $DC$  и  $AC = 2AB$ . Найдите  $\angle ABC$ .

#### **Решение**

В равнобедренном треугольнике  $ADC$  из точки  $D$  проведем медиану  $DE$  на сторону  $AC$ , которая также будет являться и высотой.

Тогда  $AE = \frac{1}{2}AC = AB$ . Треугольники  $AED$  и  $ABD$  равны по двум сторонам и углу между ними:  $AE = AB$ ,  $AD$  — общая сторона и  $\angle DAE = \angle DAB$ .

Следовательно,  $\angle ABC = \angle ABD = \angle AED = 90^\circ$ .

**Ответ:**  $90^\circ$ .

### Задача 2.3.3.4. (25 баллов)

Тема: десятичная запись натурального числа.

#### Условие

В натуральном двузначном числе  $a$  цифры поменяли местами и получили двузначное число  $b$ . Оказалось, что сумма чисел  $a$  и  $b$  делится на 5, а их разность — на 27.

Найдите все возможные значения числа  $a$ . В ответ запишите сумму всех полученных чисел.

#### Решение

Пусть  $a = \overline{xy} = 10x + y$  и  $b = \overline{yx} = 10y + x$ .

Тогда

$$a + b = 11x + y = 11(x + y).$$

Так как по условию  $a + b = 11(x + y) \div 5$  и числа 5 и 11 взаимно просты, то

$$(x + y) \div 5. \quad (2.3.1)$$

Далее из второго условия  $a - b = 9x - 9y = 9(x - y) \div 27$ , следует, что

$$(x - y) \div 3. \quad (2.3.2)$$

Осталось перебрать все возможные значения цифр  $x$  и  $y$ , удовлетворяющих условиям (2.3.1) и (2.3.2). Непосредственной проверкой можно убедиться, что этим условиям удовлетворяют пары  $(1; 4)$ ,  $(2; 8)$ ,  $(4; 1)$ ,  $(5; 5)$ ,  $(6; 9)$ ,  $(8; 2)$  и  $(9; 6)$ .

Таким образом, получаем пять чисел, сумма которых равна  $14 + 28 + 41 + 55 + 69 + 82 + 96 = 385$ .

**Ответ:** 385.

### Задача 2.3.3.5. (25 баллов)

Тема: текстовая задача.

#### Условие

Команда «Математики» за последние три года, согласно протоколам, приняла участие в 111 матчах по мини-футболу (в это число вошли и игры, которые были отменены по техническим причинам). При анализе результатов было замечено:

- сколько-то игр было выиграно;
- ничьи составляют 45% от всех игр, в которых не были одержаны победы;
- количество матчей, в которых были допущены поражения, к количеству отмененных игр относится как 1 : 2.

Какое количество матчей «Математики» проиграли?

### ***Решение***

Пусть было одержано  $x$  побед. Тогда количество игр, которые были сыграны вничью, проиграны или были отменены, равно  $111 - x$ .

Тогда  $\frac{9}{20}(111 - x)$  — количество игр, сыгранных вничью.

Найдем количество игр, которые были проиграны, или отменены:

$$(111 - x) - \frac{9}{20}(111 - x) = \frac{1221 - 11x}{20}.$$

Тогда количество игр, в которых были поражения, равно

$$y = \frac{1221 - 11x}{60} \in Z.$$

Получили диофантово уравнение

$$11x + 60y = 1221.$$

Выразим  $x$ :

$$x = 111 - 60 \cdot \frac{y}{11}.$$

Таким образом,  $y \mid 11$  и  $y > 0$ .

Рассмотрим различные случаи относительно  $y$ :

1.  $y = 11$ . Тогда  $x = 111 - 60 = 51$ .
2.  $y = 22$ . Тогда  $x = 111 - 120 = -9$ . Количество игр не может быть отрицательным числом. Следовательно, данный случай, как и все последующие, не подходит.

Таким образом, количество игр, в которых были получены поражения, равно 11.

**Ответ:** 11.

### **2.3.4. Вторая волна. Задачи 10–11 класса**

Задачи второй волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63477/enter/>.

### **Задача 2.3.4.1. (10 баллов)**

*Темы: стереометрия, центральная симметрия.*

#### **Условие**

Прямоугольный параллелепипед имеет объем, равный 30. Его рассекли на две части, проведя плоскость через точку пересечения всех трех его диагоналей.

Чему равно максимальное значение объема одной из этих двух частей?

#### **Решение**

При центральной симметрии плоскости переходят в параллельные им плоскости, а прямые — в параллельные им прямые. Диагонали параллелепипеда делятся точкой пересечения пополам, поэтому он имеет центр симметрии. При центральной симметрии любая точка параллелепипеда, не находящаяся на секущей плоскости, перейдет в точку, которая находится с другой стороны от любой плоскости, проходящей через этот центр, так как эти две точки и центр симметрии находятся на одной прямой, которая пересекает эту плоскость.

Таким образом, плоскость делит параллелепипед на две части, которые переходят друг в друга при центральной симметрии. Следовательно, их объемы должны быть равны. Это означает, что часть параллелепипеда имеет объем, равный половине объема параллелепипеда

**Ответ:** 15.

### **Задача 2.3.4.2. (15 баллов)**

*Темы: теорема Виета, неравенство о средних, многочлены.*

#### **Условие**

Путешественник достал древнюю карту спрятанных сокровищ на острове Пасхи. Путь к пещере, в которой пиратами был закопан клад, был зашифрован с помощью квадратного уравнения. К сожалению, с течением времени запись одного из коэффициентов стерлась, и поэтому путешественник не смог его точно восстановить.

Оказалось, что оно имеет следующий вид:  $x^2 + 6x + a = 0$ .

Здесь буквой  $a$  обозначен неизвестный коэффициент.

Уравнение использовалось для того, чтобы можно было разделить инструкцию по поиску сокровища на несколько частей таким образом, чтобы совершенно невозможно было бы понять, что и где искать, если хотя бы одной части недостает.

У путешественника были все части инструкции, поэтому он смог понять, что нужно от нужной точки на побережье идти ровно  $P$  км на юг вдоль единственной тропы, затем  $Q = \frac{P}{2}$  км на запад, а потом повернуться на северо-восток и идти прямо, пока вершина вулкана Теревака, кратер Рано-Арои, не станет виден под углом

ровно  $10R^\circ$  над уровнем горизонта. Рядом с этим местом и находится пещера. Здесь  $P, Q$  — корни данного квадратного уравнения, упорядоченные по возрастанию,  $R = 2P + Q$ .

Может ли путешественник, исходя из данных условий, однозначно найти два этих корня?

Если может, напишите в ответ число  $R$ . Если не может, напишите в ответ число 0.

### ***Решение***

Запишем теорему Виета для квадратного уравнения:

$$\begin{cases} P + Q = -6, \\ PQ = a. \end{cases}$$

В условии указано, что  $Q = \frac{P}{2}$ . Подставив в первое уравнение, получаем, что  $P = -4$ ,  $Q = -2$ .

**Ответ:**  $-10$ .

### ***Задача 2.3.4.3. (20 баллов)***

*Темы: арифметическая задача, симметрия.*

#### ***Условие***

Исследователи выращивают экспериментальную культуру грибов. Эти грибы размножаются почкованием. Гриб порождает два новых гриба каждые 4 ч. Только что появившийся гриб слишком маленький, и поэтому он должен еще 6 ч расти, прежде чем размножаться, таким образом, первое потомство от нового гриба возникает лишь через 10 ч после его появления из почки.

Сколько грибов, включая только что появившихся, будет в лаборатории через 28 ч, если изначально там был один гриб, который породит два новых гриба только через 4 ч.

### ***Решение***

Самый первый гриб за 28 ч успеет породить только три поколения грибов, так как для появления четвертого поколения нужно 30 ч. Поэтому чтобы ответить на вопрос задачи, нужно посчитать, сколько грибов успеют отпочковаться от грибов, которые породил первый гриб, а потом посчитать также третье поколение.

Первые два гриба, отпочковавшиеся через 4 ч, создадут еще четыре гриба в 14 ч, еще четыре — в 18 ч, еще четыре — в 22 ч и еще четыре в — 26 ч. Всего они породят 16 грибов.

Вторые два гриба, появившиеся через 8 ч, создадут еще четыре гриба в 18 ч, еще четыре — в 22 ч и еще четыре гриба — в 26 ч. Всего они породят 12 грибов.

Трети два гриба, появившиеся через 12 ч, создадут еще четыре гриба в 22 ч, и еще четыре гриба — в 26 ч. Всего они породят восемь грибов.

Четвертые два гриба, появившиеся через 16 ч, создадут еще четыре гриба в 26 ч.

Пятые два гриба — в 20 ч, шестые два гриба — в 24 ч, а седьмые два гриба в 28 ч не успеют породить никаких новых грибов — это еще шесть грибов.

Таким образом, можно посчитать количество грибов первого и второго поколения:

$$N_1 = 7 \cdot 2 = 14;$$

$$N_2 = 16 + 12 + 8 + 4 = 40.$$

Осталось посчитать третье поколение. Оно образуется в 24 ч и 28 ч из первых четырех грибов из первых двух грибов, в 28 ч из вторых четырех грибов из первых двух грибов и в 28 ч из первых четырех грибов из вторых двух грибов. То есть еще восемь грибов:

$$N_3 = 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 = 32.$$

Суммарно получаем:

$$N = 1 + N_1 + N_2 + N_3 = 1 + 14 + 40 + 32 = 87.$$

**Ответ:** 87.

### Задача 2.3.4.4. (25 баллов)

Темы: прямоугольный треугольник, теорема Пифагора, теорема косинусов.

#### Условие

В прямоугольном равнобедренном треугольнике  $ABC$  с прямым углом  $C$  проведена биссектриса  $AL$ . Из точки  $L$  к стороне  $BC$  проведен перпендикуляр, который пересек сторону  $AB$  в точке  $M$ . Перпендикуляр, построенный к стороне  $AB$  в точке  $M$ , пересекает сторону  $AC$  в точке  $N$ .

Чему равен угол  $ANL$ ? Ответ приведите в градусах.

#### Решение

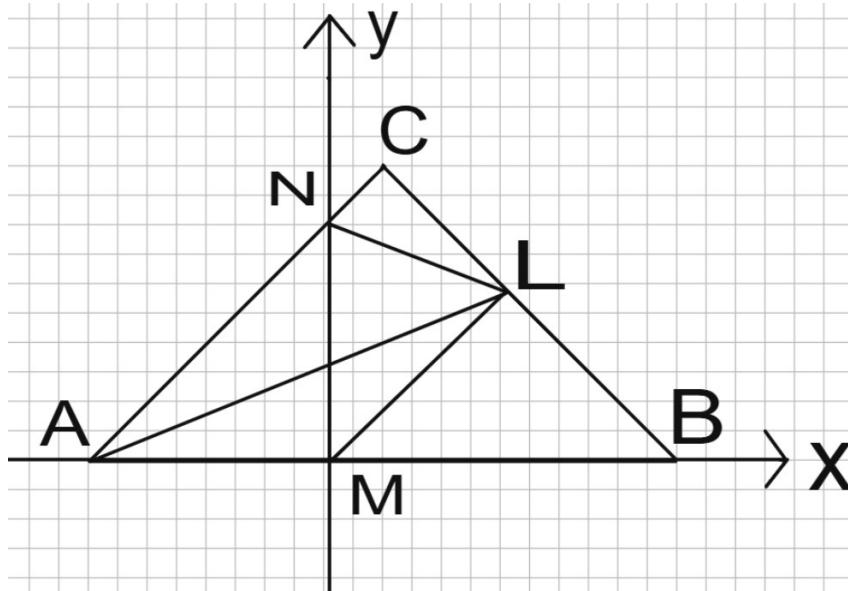
$MN = AM$ , значит, угол  $ANM$  тоже равен  $45^\circ$  и  $NM$  перпендикулярно  $AB$ .

Тогда углы  $NML$  и  $BML$  тоже равны по  $45^\circ$ .

Пусть  $AM = 1$  (в условии никаких длин нет, поэтому можем за единицу длины взять любой отрезок). Тогда

$$AN = \sqrt{2};$$

$$AL = 2 \cdot AM \cdot \cos 22,5^\circ = 2 \cdot \sqrt{\frac{\sqrt{2}+2}{4}} = \sqrt{\sqrt{2}+2}.$$



Чтобы найти  $NL$ , используем метод координат. Проведем горизонтальную ось через  $AB$ , вертикальную ось через  $MN$ . Тогда точка  $N$  имеет координаты  $(0; 1)$ . Что же касается точки  $L$ , то ее координаты  $x$  и  $y$  совпадают, а длина  $ML$  равна 1. Следовательно, они равны  $L\left(\frac{\sqrt{2}}{2}; \frac{\sqrt{2}}{2}\right)$ .

Используя формулу расстояния между двумя точками, получаем:

$$NL^2 = \frac{1}{2} + \left(1 - \frac{\sqrt{2}}{2}\right)^2 = 2 - \sqrt{2}.$$

Обозначив угол  $ALN$  за  $x$ , применим теорему косинусов:

$$2 = 2 - \sqrt{2} + \sqrt{2} + 2 - 2 \cdot \sqrt{\sqrt{2}+2} \cdot \sqrt{2-\sqrt{2}} \cdot \cos x.$$

Отсюда получаем, что:

$$\cos x = \frac{\sqrt{2}}{2}, \quad x = 45^\circ.$$

Тогда угол  $ANL$  равен  $180^\circ$  минус угол  $ALN$  и угол  $NAL$ :

$$ANL = 180 - 45 - 22,5 = 112,5.$$

**Ответ:** 112,5.

### Задача 2.3.4.5. (30 баллов)

Темы: уравнение параболы, уравнение касательной, угловой коэффициент наклона прямой.

### **Условие**

Для разработки оптической системы на основе параболических отражателей света потребовалось исследовать оптические свойства парабол. Пусть парабола задана уравнением  $y = 16x^2$ . Требуется на плоскости найти такую точку  $O$ , что все проекции этой точки на касательные к параболе лежат на оси абсцисс. Найдите координаты точки  $O$  и запишите их в ответ.

Уравнение касательной прямой к параболе (в заданной точке  $(x_0, y_0)$ ) однозначно устанавливается как уравнение невертикальной прямой, проходящей через  $(x_0, y_0)$  и имеющей единственную точку пересечения с параболой.

### **Решение**

Рассмотрим точку с абсциссой  $x_0$  на параболе. Уравнение прямой, проходящей через эту точку, в общем виде имеет вид:

$$y = a \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Приравняем его к уравнению параболы и найдем, при каком значении  $a$  они будут иметь ровно одну точку пересечения:

$$\begin{aligned} 16 \cdot x^2 &= a \cdot (x - x_0) + 16 \cdot (x_0)^2; \\ 16 \cdot x^2 - a \cdot x + x_0 \cdot a - 16 \cdot (x_0)^2 &= 0; \\ D = a^2 - 4 \cdot 16 \cdot (x_0 \cdot a - 16 \cdot x_0^2) &= (a - 32 \cdot x_0)^2 = 0; \\ a &= 32 \cdot x_0. \end{aligned}$$

Итак, запишем уравнение касательной в этой точке к параболе в виде

$$y = 32 \cdot x_0 \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Эта прямая пересечет ось абсцисс в точке с координатой  $x_1 = \frac{x_0}{2}$ .

Уравнение прямой, проходящей через эту точку перпендикулярно касательной:

$$y = -\frac{2 \cdot x - x_0}{64 \cdot x_0}.$$

Эта прямая пересечет ось ординат в точке с координатами  $(0; 0,015625)$ . Координаты этой точки не зависят от значения  $x_0$ , а значит, все такие прямые пройдут через эту точку.

**Ответ:**  $(0; 0,015625)$ .

### **2.3.5. Третья волна. Задачи 8–9 класса**

Задачи третьей волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Конкурс: <https://contest.yandex.ru/contest/63461/enter/>.

**Задача 2.3.5.1. (15 баллов)***Тема: текстовая задача.****Условие***

Начинающий предприниматель Петров закупил 1 000 единиц некоторого товара и попытался его продать с наценкой 20% за единицу продукции. Однако ожидания предпринимателя не совпали с реальностью, и он смог продать только 40% от своего объема, после чего вынужден был снизить цену на товар на 10%. В результате снижения единица товара стала стоить 5 832 руб. за штуку.

Какую чистую прибыль, то есть разность между деньгами, полученными за продажу товара и затратами на его закупку, получил Петров?

***Решение***

Пусть  $x$  руб. — цена за единицу товара, по которой совершена закупка предпринимателем Петровым. Тогда он первоначально планировал осуществить продажи по цене

$$x + 0,2x = 1,2x.$$

После снижения же цены товар стал стоить

$$1,2x - 0,1 \cdot 1,2x = 1,08x.$$

Так как известно, что после снижения единица товара стала стоить 5 832 руб. за штуку, то

$$1,08x = 5\,832 \Rightarrow x = 5\,400.$$

Таким образом, товар был закуплен 5 400 руб. за штуку, и общие затраты на его покупку составили 5 400 000 руб.

Согласно условию задачи 400 единиц товара было продано по цене  $1,2 \cdot 5\,400 = 6\,480$  руб., и всего было получено за них  $6\,480 \cdot 400 = 2\,592\,000$  руб.

Оставшиеся же 600 единиц были проданы по цене 5 832 руб. и получено за них  $5\,832 \cdot 600 = 3\,499\,200$  руб.

Тогда чистая прибыль предпринимателя Петрова будет равна

$$2\,592\,000 + 3\,499\,200 - 5\,400\,000 = 691\,200 \text{ руб.}$$

**Ответ:** 691 200.

**Задача 2.3.5.2. (15 баллов)***Тема: кombinatorika.****Условие***

Сколько существует нечетных пятизначных чисел, в которых есть хотя бы одна цифра 5?

### ***Решение***

Для того чтобы найти количество требуемых чисел, достаточно из общего количества пятизначных нечетных чисел вычесть количество чисел, в которых отсутствует цифра 5.

В десятичной записи нечетного пятизначного числа на последнюю позицию претендует пять вариантов (цифры 1, 3, 5, 7 и 9), на первую — девять вариантов (все цифры, кроме нуля), а на все остальные позиции — по 10 вариантов. Тогда общее количество пятизначных нечетных чисел будет равно

$$9 \cdot 10 \cdot 10 \cdot 10 \cdot 5 = 45\,000.$$

Для записи нечетного пятизначного числа, в десятичной записи которого отсутствует цифра 5, на каждую соответствующую позицию будет на один вариант меньше, тогда общее количество таких чисел будет равно

$$8 \cdot 9 \cdot 9 \cdot 9 \cdot 4 = 23\,328.$$

Тогда количество пятизначных нечетных чисел, в которых присутствует хотя бы одна цифра 5, равно

$$45\,000 - 23\,328 = 21\,672.$$

**Ответ:** 21 672.

### ***Задача 2.3.5.3. (20 баллов)***

*Темы: алгебра, система уравнений.*

#### ***Условие***

Наблюдательный Витя для некоторых двух различных чисел заметил интересную особенность: первое число, увеличенное на 4, будет равно квадрату второго числа, уменьшенного на 2; и наоборот, если ко второму числу прибавить 4, то результат будет равен квадрату первого числа, уменьшенного на 2. Найдите сумму квадратов данных двух чисел.

### ***Решение***

Пусть  $x, y$  — два исходных различных числа. Тогда согласно условиям задачи будем иметь систему уравнений:

$$\begin{cases} x + 4 = (y - 2)^2, \\ y + 4 = (x - 2)^2. \end{cases}$$

Вычитая из первого равенства второе, получим:

$$x - y = (y - 2)^2 - (x - 2)^2 = (y - x)(x + y - 4).$$

Так как числа  $x, y$  различны, то отсюда получаем, что  $x + y = 3$ .

Складывая же уравнения полученной системы, получим

$$x + y + 8 = (y - 2)^2 + (x - 2)^2 = y^2 - 4y + 4 + x^2 - 4x + 4.$$

Из последнего равенства получаем, что

$$x^2 + y^2 = 5(x + y) = 15.$$

**Ответ:** 15.

### Задача 2.3.5.4. (25 баллов)

*Темы: теория чисел, остатки.*

#### Условие

Петя записал на доске три числа 391, 604, 888 и задумчиво сказал Васе: «Если я сейчас эти три числа разделяю на одно и то же натуральное число, отличное от единицы, то в результате получу один и тот же остаток».

На какое натуральное число Петя планирует произвести деление исходных чисел?

#### Решение

Обозначим число, на которое производится деление, через  $x$ , а остаток через  $y$ .

Тогда каждое из записанных Петей чисел можно представить в виде:

$$\begin{aligned} 391 &= xm + y, \\ 604 &= xk + y, \\ 888 &= xn + y, \end{aligned}$$

где  $m, k$  и  $n$  — неполные частные, возникающие при делении.

Вычитая из третьего равенства второе, а из второго — первое, получим:

$$\begin{aligned} 284 &= x(n - k), \\ 213 &= x(k - m). \end{aligned}$$

Вычтем из верхнего равенства нижнее:

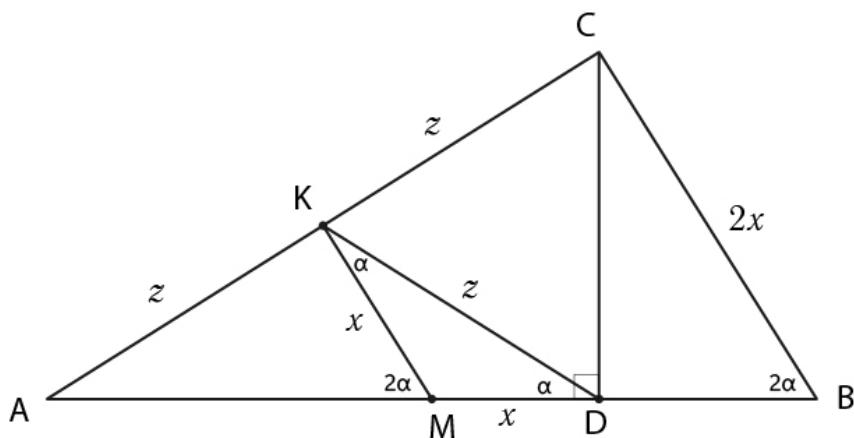
$$71 = x(n - 2k + m).$$

Так как 71 — это простое число, то  $71 = 71 \cdot 1$ , и, по условию задачи  $x \neq 1$ , то единственный возможный вариант для делителя Пети равен 71.

**Ответ:** 71.

**Задача 2.3.5.5. (25 баллов)***Тема: планиметрия.***Условие**

$CD$  — высота остроугольного треугольника  $ABC$ ,  $M$  — середина стороны  $AB$  и  $\angle ABC = 2\angle BAC$ . Найдите отношение  $BC : MD$ .

**Решение**

На стороне  $AC$  отметим ее середину — точку  $K$ .

Тогда  $AK = KC$  и  $AM = MB$  (по условию задачи), следовательно,  $MK$  — средняя линия треугольника  $ABC$  и  $BC = 2MK$ .

Докажем, что  $MK = MD$ .

По свойству медианы прямоугольного треугольника, проведенной из вершины прямого угла, в треугольнике  $ADC$ :  $DK = \frac{1}{2}AC = AK$ .

Таким образом, треугольник  $AKD$  — равнобедренный и  $\angle KAD = \angle KDA$  как углы при основании  $KD$ .

Так как  $MK$  — средняя линия треугольника  $ABC$ , то  $MK \parallel BC$  и  $\angle AMK = \angle ABC = 2\angle BAC = 2\angle KAD = 2\angle KDA = 2\angle KDM$ .

По теореме о внешнем угле для треугольника  $MKD$

$$\angle AMK = \angle KDM + \angle MKD.$$

Тогда из последних двух равенств следует, что  $\angle KDM = \angle MKD$  и треугольник  $MKD$  — равнобедренный.

Следовательно,  $MK = MD$ , и так как  $BC = 2MK = 2MD$ , то  $BC : MD = 2 : 1$ .

**Ответ:** 2.

## 2.3.6. Третья волна. Задачи 10–11 класса

Задачи третьей волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63478/enter/>.

### Задача 2.3.6.1. (10 баллов)

Темы: осевая симметрия, равнобедренный треугольник, движение.

#### Условие

Известно, что выпуклая фигура  $\Phi$  на плоскости устроена таким образом, что она симметрична относительно любой прямой, которая проходит через точку  $O$  на этой плоскости. Самое большое расстояние между двумя точками, принадлежащими фигуре  $\Phi$ , равно дроби, в числителе которой шесть, а в знаменателе квадратный корень из числа  $\pi$ .

Чему равна площадь фигуры  $\Phi$ ?

#### Решение

Заметим, что такой фигурой может быть только круг. Можно это доказать и более строго.

Возьмем на фигуре точку  $A$ , максимально удаленную от точки  $O$ , и проведем прямую  $OA$ . Затем будем плавно поворачивать эту прямую вокруг точки  $O$  и смотреть образы точки  $A$ , которые получаются при осевой симметрии относительно прямой. Все они должны лежать внутри фигуры. Расстояние до этих точек  $X$  от  $O$  всегда будет одним и тем же, поэтому они образуют окружность. Таким образом, фигура  $\Phi$  содержит эту окружность и не выходит за ее пределы. Аналогично, рассмотрев любые другие точки на фигуре, получаем, что фигура  $\Phi$  содержит все точки, лежащие внутри этой окружности.

Следовательно,  $\Phi$  является кругом радиуса 3, деленного на корень из числа  $\pi$ . Его площадь равна 9.

**Ответ:** 9.

### Задача 2.3.6.2. (15 баллов)

Темы: составление уравнений, составление пропорций, проценты.

#### Условие

Находясь на борту космического корабля, главный двигатель за первый час израсходовал 40% всего запаса анобтаниума, а вспомогательные двигатели вместе за это же время израсходовали лишь 300 г анобтаниума. За следующий час главный

двигатель израсходовал 80% оставшегося топлива, а вспомогательные двигатели израсходовали 100 г топлива на двоих. В итоге на борту корабля осталось 800 г топлива. Сколько килограммов фантастического топлива было на борту до начала полета?

### ***Решение***

Найдем массу анобтаниума, оставшегося к концу первого часа.

Не было израсходовано главным двигателем к этому моменту  $100 + 800 = 900$  г.

Это составляет  $100 - 80 = 20\%$ .

Составим пропорцию и решим ее:

$$\begin{aligned} 20\% &= 900, \\ 100\% &= ? \end{aligned}$$

Значит, к концу первого часа оставалось  $900 : 0,2 = 4500$  г анобтаниума.

Найдем массу топлива к началу первого часа.

Не было израсходовано главным двигателем к этому моменту  $4500 + 300 = 4800$  г, что составляет  $100 - 40 = 60\%$ .

Составим пропорцию и решим ее:

$$\begin{aligned} 60\% &= 4800, \\ 100\% &= ? \end{aligned}$$

Значит, к началу первого часа было  $4800 : 0,6 = 8000$  г, что составляет 8 кг.

**Ответ:** 8.

### ***Задача 2.3.6.3. (20 баллов)***

*Темы: уравнение параболы, уравнение прямой.*

### ***Условие***

Известно, что три различные точки  $A(2; 4)$ ,  $B(x; 6)$ ,  $C(6; y)$  расположены на координатной плоскости таким образом, что через них нельзя провести параболу с вертикальной осью. При этом также известно, что  $x$  — минимальное натуральное подходящее число, неравное единице.

Найдите величину  $x + y$ .

### ***Решение***

Через три точки нельзя провести параболу тогда и только тогда, когда они расположены на одной прямой. Действительно, прямая не может пересекать параболу в трех точках, так как квадратное уравнение имеет не больше двух корней. С другой стороны, если три точки не лежат на одной прямой, то через них всегда можно провести параболу. Покажем это.

Пусть на числовой прямой есть три точки с координатами  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ . Запишем уравнение параболы в следующем виде:

$$y = y_1 \frac{(x - x_2) \cdot (x - x_3)}{(x_1 - x_2) \cdot (x_1 - x_3)} + y_2 \frac{(x - x_1) \cdot (x - x_3)}{(x_2 - x_1) \cdot (x_2 - x_3)} + y_3 \frac{(x - x_1) \cdot (x - x_2)}{(x_3 - x_1) \cdot (x_3 - x_2)}.$$

Первое слагаемое равно нулю во второй и третьей точке, и равно  $y_1$  в первой, аналогичным образом устроены второе и третье слагаемые, так что это уравнение задает функцию, проходящую через три точки. Однако надо еще проверить, что уравнение задает именно параболу. Для этого нужно, чтобы коэффициент при  $x^2$  не равнялся нулю.

$$\begin{aligned} \frac{y_1}{(x_1 - x_2) \cdot (x_1 - x_3)} + \frac{y_2}{(x_2 - x_1) \cdot (x_2 - x_3)} + \frac{y_3}{(x_3 - x_1) \cdot (x_3 - x_2)} &\neq 0; \\ y_1 \cdot (x_2 - x_3) - y_2 \cdot (x_1 - x_3) + y_3 \cdot (x_1 - x_2) &\neq 0. \end{aligned}$$

Можно убедиться, что это условие означает, что три точки не лежат на одной прямой. А именно, нахождение трех точек на одной прямой можно записать следующим образом:

$$\begin{aligned} \frac{y_1 - y_2}{x_1 - x_2} &= \frac{y_1 - y_3}{x_1 - x_3}; \\ (y_1 - y_2) \cdot (x_1 - x_3) &= (y_1 - y_3) \cdot (x_1 - x_2); \\ y_1 \cdot (x_1 - x_3) - y_1 \cdot (x_1 - x_2) &= y_2 \cdot (x_1 - x_3) - y_3 \cdot (x_1 - x_2); \\ y_1 \cdot (x_2 - x_3) + y_3 \cdot (x_1 - x_2) - y_2 \cdot (x_1 - x_3) &= 0. \end{aligned}$$

Таким образом, если это условие выполнено, то через три точки проходит прямая, и не проходит никакая парабола. А если оно не выполнено, то проходит единственная парабола, и нельзя провести никакую прямую.

Тогда выразим угловой коэффициент этой прямой тремя разными способами:

$$k = \frac{2}{x - 2} = \frac{y - 6}{6 - x} = \frac{y - 4}{4}.$$

Отсюда получаем, что

$$y = \frac{4 \cdot x}{x - 2}.$$

Минимальное натуральное  $x$ , не равное единице, которое подходит — это  $x = 3$ . Тогда  $y = 12$ .

Значит,  $x + y = 15$ .

**Ответ:** 15.

### Задача 2.3.6.4. (25 баллов)

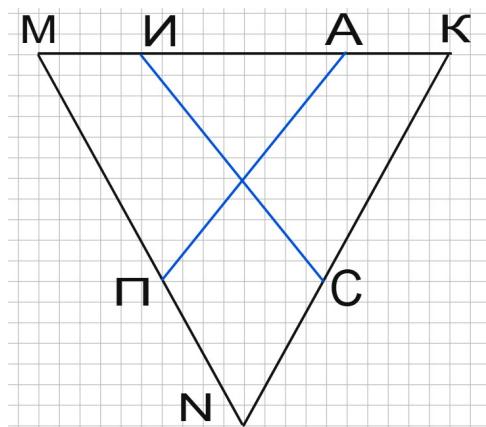
Темы: равносторонний треугольник, первый признак равенства треугольников.

### **Условие**

Три прямые дороги образуют треугольник с равными сторонами, длина которых равна 1 000 м. У этих дорог стоят четыре человека, каждый на обочине одной из трех дорог. Иван и Александр стоят возле одной дороги в 500 м друг от друга. Сергей и Петр стоят у обочин двух других дорог. Сергею идти до Ивана 1 500 м по дорогам кратчайшим путем, Александру до Петра тоже. Между дорогами расположено поле. Какая величина получится, если к расстоянию от Сергея до Ивана по прямой (то есть по полю, а не по дорогам) добавить половину расстояния от Ивана до Александра вдоль дороги, возле которой они стоят, и вычесть расстояние от Александра до Петра по прямой (по полю)?

### **Решение**

Нарисуем расположение всех этих четырех человек. Расположение Сергея и Петра здесь определяется из того условия, что путь до Ивана и Александра соответственно должен занимать 1 500 м, в то время как расстояние по одной стороне не больше 1 000 м, а по другой не больше 500 м.



Используя указанные расстояния, можем записать:

$$MA + MP = KI + KC, \text{ а значит, } MI + MP = KA + KC = 1000 \text{ м.}$$

$$MI + KA = IA = 500 \text{ м. Кроме того, длина } KM \text{ равна } 1000 \text{ м.}$$

Отсюда выходит, что  $KA = 1000 - KC = 1000 - MA$ , а значит,  $KC = MA$ . Аналогично выходит, что  $KI = MP$ .

Тогда треугольники  $KIC$  и  $MPA$  равны друг другу по двум сторонам и углу между ними.

$$\text{Следовательно, } AP = CI, \text{ а значит, } AP - CI + 0,5 \cdot IA = 250 \text{ м.}$$

**Ответ:** 250.

### **Задача 2.3.6.5. (30 баллов)**

Темы: делители числа, произведение делителей, разложение на множители.

### ***Условие***

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей (рассматриваются все делители, включая само число и единицу). Третья часть всех делителей не делится на 3. Половина четных делителей делится на 5. Само число при этом не превосходит 10 000. Напишите в ответ максимальное число, которое подходит под этим условия.

### ***Решение***

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей.

Это значит, что оно делится на  $2^5$  степени, но не делится на  $2^6$ .

Третья часть всех делителей не делится на 3.

Это значит, что оно делится на  $3^2$ , но не делится на  $3^3$ .

Половина четных делителей делится на 5.

Это значит, что оно делится на 5, но не делится на 25.

Если перемножим  $2^5$  на  $3^2$  и на 5, то получим 1440. Минимальное число, подходящее под условия выше, но большее этого числа, равно  $7 \cdot 1440 = 10\,080 > 10\,000$ .

Следовательно, под все условия подходит только число 1440.

**Ответ:** 1440.

### **2.3.7. Четвертая волна. Задачи 8–9 класса**

Задачи четвертой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63462/enter/>.

#### ***Задача 2.3.7.1. (15 баллов)***

*Темы:* теория чисел, признаки делимости.

### ***Условие***

На доске записано число 202420252026. Танечка хочет убрать несколько цифр из исходного числа так, чтобы получившийся результат делился на 45 и являлся наибольшим из всех возможных. Какое число запишет на доске Танечка?

### ***Решение***

Для того чтобы число Танечки было бы кратно 45, необходимо выполнение условий делимости на 5 и 9. Следовательно, число должно заканчиваться на 0 или

на 5. В данном случае первым делом Танечка должна убрать последние две цифры и получить 2024202520.

Для выполнения условия делимости на 9 необходимо, чтобы сумма цифр числа была бы 9. Сумма цифр сейчас равна 19. Ближайшая сумма, кратная 9, равна 18, но 1 в числе нет, следовательно, следующий вариант — 9. Для этого из оставшегося числа ей нужно вычеркнуть цифры, дающие в сумме 10. Тогда наибольшее число, которое может получить Танечка, — 202050.

**Ответ:** 202050.

### **Задача 2.3.7.2. (15 баллов)**

*Тема: десятичная запись натурального числа.*

#### **Условие**

Найдите все трехзначные натуральные числа  $\overline{abc}$ , удовлетворяющие условию

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca}.$$

В ответ запишите сумму всех найденных чисел.

#### **Решение**

Распишем равенство, заданное в условии задачи

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca};$$

$$100a + 10b + c = 10a + b + 10b + c + 10c + a;$$

$$100a + 10b + c = 11a + 11b + 11c;$$

$$89a = 10c + b.$$

Так как  $a$ ,  $b$ ,  $c$  — цифры, то единственным решением данного уравнения является набор  $a = 1$ ,  $b = 9$ ,  $c = 8$ . Следовательно, единственное число, удовлетворяющее условию задачи, это 198.

**Ответ:** 198.

### **Задача 2.3.7.3. (20 баллов)**

*Темы: алгебра, квадратный трехчлен.*

#### **Условие**

Найдите количество значений параметра  $b$ , при которых все корни уравнения  $x^2 + bx + 2026 = 0$  целые.

***Решение***

Пусть  $x_1$  и  $x_2$  — целые корни данного уравнения. Тогда согласно теореме Виета:

$$x_1 \cdot x_2 = 2026.$$

Так как 2026 раскладывается на множители

$$2026 = 1 \cdot 2026 = 2 \cdot 1013,$$

то получаем четыре набора для значений корней

$$(1; 2026), (-1; -2026), (2; 1013), (-2; -1013).$$

Зная значения корней, также по теореме Виета найдем значения параметра  $b$ :

$$b = -(x_1 + x_2).$$

Таким образом, всего существует четыре значения параметра  $b = \{-2027; 2027; -2015; 2015\}$ , при каждом из которых уравнение имеет целые корни.

**Ответ:** 4.

***Задача 2.3.7.4. (25 баллов)***

*Тема: геометрическая вероятность.*

***Условие***

В треугольнике  $ABC$  на биссектрисе  $BD$  отмечена точка  $E$  так, что  $BE = ED$ . Найти вероятность, что точка, брошенная в треугольник  $ABC$ , попадет в треугольник  $AED$ , если  $AB = 3$  и  $BC = 5$ .

Ответ выразите в долях и при необходимости округлите его до четвертого знака после запятой.

***Решение***

Согласно определению геометрической вероятности, требуемая вероятность будет равна отношению площадей треугольников  $AED$  и  $ABC$ .  $AE$  — медиана треугольника  $ABE$ , следовательно,  $S_{ABD} = 2S_{AED}$ .

Площади треугольников  $ABD$  и  $BDC$  относятся как длины их оснований  $AD$  и  $DC$ , то есть

$$\frac{S_{ABD}}{S_{BDC}} = \frac{AD}{DC} = \frac{AB}{BC} = \frac{3}{5}.$$

Последнее равенство выполняется согласно свойству биссектрисы  $BD$  в треугольнике  $ABC$ . Тогда

$$S_{ABC} = S_{ABD} + S_{BDC} = S_{ABD} + \frac{5}{3}S_{ABD} = \frac{8}{3}S_{ABD} = \frac{16}{3}S_{AED}.$$

Из последнего равенства следует отношение

$$\frac{S_{AED}}{S_{ABC}} = \frac{3}{16} = 0,1875.$$

Таким образом, вероятность того, что точка брошенная в треугольник  $ABC$ , попадет в треугольник  $AED$ , равна 0,1875.

**Ответ:** 0,1875.

### Задача 2.3.7.5. (25 баллов)

*Тема: алгебра.*

#### *Условие*

При каком значении числа  $a$  сумма квадратов чисел  $x$  и  $y$  будет принимать наибольшее значение, если известно, что сумма этих чисел равна  $2a + 1$ , а произведение равно  $4a^2 + 8a - 4$ ?

#### *Решение*

По условию задачи  $x + y = 2a + 1$  и  $xy = 4a^2 + 8a - 4$ .

Воспользуемся формулой квадрата суммы двух чисел

$$(x + y)^2 = x^2 + 2xy + y^2.$$

Отсюда

$$\begin{aligned} x^2 + y^2 &= (x + y)^2 - 2xy = (2a + 1)^2 - 2(4a^2 + 8a - 4) = 4a^2 + 4a + 1 - 8a^2 - 16a + 8 = \\ &= -4a^2 - 12a + 9 = -(4a^2 + 12a + 9) + 18 = -(2a + 3)^2 + 18. \end{aligned}$$

В полученном выражении первое слагаемое принимает неположительные значения при любом  $a$ . Следовательно, сумма квадратов чисел  $x$  и  $y$  будет максимальной при  $2a + 3 = 0$  или  $a = -1,5$ .

Проверим, что при данном значении параметрам  $a = -1,5$  числа  $x$  и  $y$  действительно существуют. В этом случае  $x + y = -2$  и  $xy = -7$ .

Выразив из первого равенства  $y = -x - 2$  и подставив его во второе, после преобразований получим уравнение  $x^2 + 2x - 7 = 0$ . Дискриминант данного уравнения равен 32, следовательно, корни уравнения существуют, по которым однозначным образом восстанавливаются решения построенной системы. Откуда и следует существования чисел  $x$  и  $y$ , заданных в условии задачи.

**Ответ:**  $-1,5$ .

## 2.3.8. Четвертая волна. Задачи 10–11 класса

Задачи четвертой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/63479/enter/>.

### Задача 2.3.8.1. (10 баллов)

Темы: кратчайший путь, параллельный перенос.

#### Условие

Склад находится в месте, отмеченном на карте точкой  $A$ . Нужно проложить дорогу до берега реки, затем построить мост, перпендикулярный течению реки, и от другого берега проложить дорогу до деревни, отмеченной на карте точкой  $B$ . Пример подобного построения на рисунке.

Берега реки здесь нарисованы как параллельные прямые. Координатная ось  $Ox$  на рисунке отсчитывает положения моста относительно реки в километрах. В примере, приведенном на рисунке, мост проходит через метку 1 км.

Через какую метку должен проходить мост, чтобы сумма длин пути от склада  $A$  до реки по дороге и от противоположного берега реки до деревни  $B$  была наименьшей? Ответ дайте в километрах.

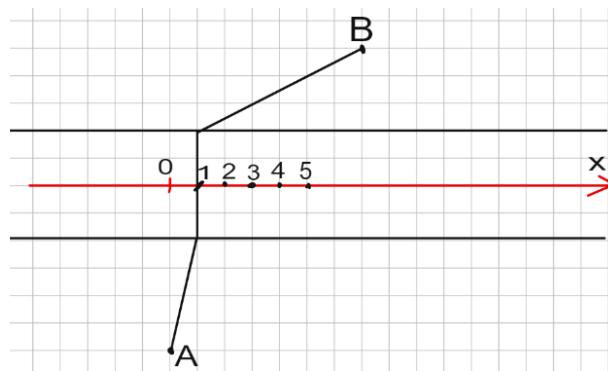


Рис. 2.3.1

#### Решение

Если вырезать с карты реку и соединить точки  $A$  и  $B$  прямой, то это и будет кратчайший путь, их соединяющий. Чтобы получить путь до реки, нужно после этого вновь вставить реку. Продемонстрируем эти операции с помощью рис. 2.3.2–2.3.3.

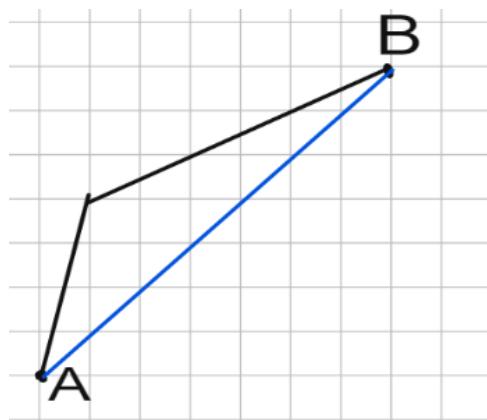


Рис. 2.3.2

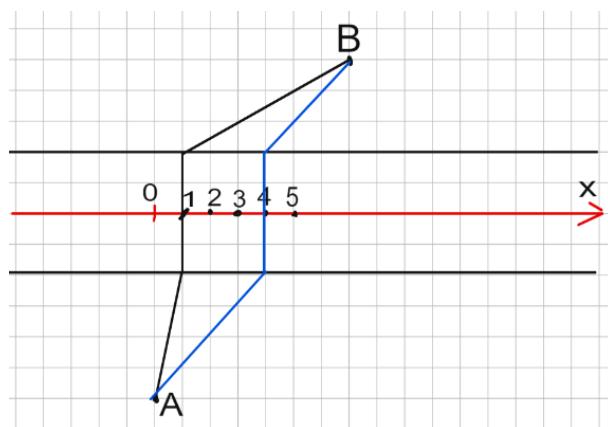


Рис. 2.3.3

**Ответ:** 4.

### Задача 2.3.8.2. (15 баллов)

Темы: составление уравнений, составление пропорций, решение уравнений.

#### Условие

Два космических корабля стартуют одновременно с одной планеты и направляются к Альфе Центавра, расстояние до которой составляет 4,37 св. лет. Один корабль движется со скоростью 0,1 св. год в год, а другой — со скоростью 0,2 св. год в год.

Через сколько лет расстояние до Альфы Центавра для более быстрого корабля будет в три раза меньше, чем для более медленного корабля? Ответ приведите с точностью до сотых.

#### Решение

Обозначим время, прошедшее с начала пути, как  $t$  лет.

Расстояние, пройденное медленным кораблем, равно  $0,1t$  св. год, и оставшееся расстояние до Альфы Центавра для медленного корабля  $4,37 - 0,1t$  св. год.

Расстояние, пройденное быстрым кораблем, равно  $0,2t$  св. год, и оставшееся расстояние до Альфы Центавра для быстрого корабля  $4,37 - 0,2t$  св. год.

По условию задачи, остаток пути для быстрого корабля в 3 раза меньше, чем остаток пути для медленного корабля:

$$4,37 - 0,2t = \frac{1}{3}(4,37 - 0,1t).$$

Умножим обе стороны на 3:

$$3(4,37 - 0,2t) = 4,37 - 0,1t;$$

$$13,11 - 0,6t = 4,37 - 0,1t.$$

Переносим все  $t$  в одну сторону и постоянные в другую:

$$13,11 - 4,37 = 0,6t - 0,1t;$$

$$8,74 = 0,5t.$$

Делим обе стороны на 0,5:

$$t = \frac{8,74}{0,5} = 17,48.$$

Таким образом, искомое время равно 17,48 лет.

**Ответ:** 17,48.

### Задача 2.3.8.3. (20 баллов)

Темы: квадратный трехчлен, функции, неопределенные коэффициенты.

#### Условие

Функция  $f(x)$  является квадратным трехчленом и может быть описана следующим образом:

$$f(x) = (f(1) + f(-1) + f(0)) \cdot x^2 + (f(1) + 2 \cdot f(0)) \cdot x - 1.$$

В то же время квадратный трехчлен в общем виде может быть записан так:

$$f(x) = a \cdot x^2 + b \cdot x + c.$$

Найдите минимальное значение величины  $a^2 + 2b^2 + 3c^2$  при данных условиях.

***Решение***

Подставим  $f(x)$  в общем виде в первую формулу из условия:

$$a \cdot x^2 + b \cdot x + c = (a + b + c + a - b + c + c) \cdot x^2 + (a + b + c + 2 \cdot c) \cdot x - 1;$$

$$\begin{cases} a = 2 \cdot a + 3 \cdot c, \\ b = a + b + 3 \cdot c, \\ c = -1. \end{cases}$$

Отсюда получаем, что  $a = 3$ ,  $c = -1$ . Чтобы искомая величина была минимальной, нужно, чтобы коэффициент  $b = 0$ .

**Ответ:** 12.

***Задача 2.3.8.4. (25 баллов)***

*Темы: делители числа, произведение делителей, разложение на множители.*

***Условие***

Произведение всех делителей числа 1 000, включая само это число и единицу, равно  $10^k$ .

Чему равно  $k$ ?

***Решение***

$$1\ 000 = 2^3 \cdot 5^3.$$

Комбинируя все возможные способы выбрать степень двойки и степень пятерки, входящие в делитель, получаем все  $(3+1) \cdot (3+1) = 16$  вариантов, каждый из которых соответствует делителю числа. При этом эти 16 делителей можно разбить на пары, произведение в каждой дает 1 000:

$$1\ 000 = 1 \cdot 1\ 000 = 2 \cdot 500 = 4 \cdot 250 = 8 \cdot 125 = 5 \cdot 200 = 10 \cdot 100 = 20 \cdot 50 = 25 \cdot 40.$$

Тогда выходит, что это будет число  $1\ 000^8$ , а значит,  $10^{24}$ .

**Ответ:** 24.

***Задача 2.3.8.5. (30 баллов)***

*Темы: равносторонний треугольник, первый признак равенства треугольников.*

**Условие**

Дан квадрат  $ABCD$ . На сторонах  $CB$  и  $CD$  отмечены точки  $L$  и  $K$  соответственно такие, что  $CL = CK$ . Из точки  $C$  на отрезок  $LD$  опущен перпендикуляр в точку  $E$ .

Пусть  $AE = 60$ ,  $EK = 91$ . Найдите длину  $AK$ .

**Решение**

Сделаем рис. 2.3.4.

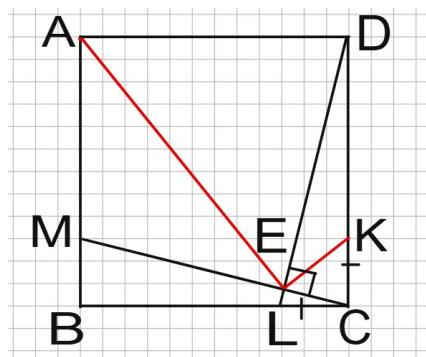


Рис. 2.3.4

Одно из возможных решений заключается в использовании метода координат. Обозначим длину квадрата за единицу, а  $CL = LK = x$ .

Тогда

$$L(1-x; 0); \quad K(1; x); \quad \overrightarrow{LD} = (x; 1); \quad \overrightarrow{CE} = t \cdot (1; -x); \quad E(1+t; -x \cdot t); \quad \overrightarrow{LE} = (t+x; -x \cdot t).$$

Так как вектора  $LE$  и  $LD$  должны быть сонаправлены, то

$$\frac{t+x}{x} = -x \cdot t; \quad t = -\frac{x}{1+x^2}; \quad E\left(1 - \frac{x}{1+x^2}; 1 - \frac{1}{1+x^2}\right);$$

$$\overrightarrow{AE} = \left(1 - \frac{x}{1+x^2}; -\frac{1}{1+x^2}\right); \quad \overrightarrow{EK} = \left(\frac{x}{1+x^2}; x - \frac{x^2}{1+x^2}\right).$$

Посчитаем скалярное произведение:  $\overrightarrow{AE} \cdot \overrightarrow{EK} = 0$ . Это значит, что треугольник  $AEK$  прямоугольный.

Тогда  $AK$  можно найти по теореме Пифагора:  $60^2 + 91^2 = 109^2$ .

**Ответ:** 109.

## 2.4. Инженерный тур

Задачи первого этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/66696/enter/>.

### ***Задача 2.4.1. Доверяй, но проверяй! (5 баллов)***

*Тема: анализ игр.*

#### ***Условие***

Ленивый геймдизайнер решил воспользоваться ИИ, чтобы проанализировать жанры популярных игр прошлого года. Однако, пока писал запрос, допустил ошибку, и в итоге ИИ дал каждой игре по одному ошибочному жанру. Проанализируйте игры и найдите один неподходящий жанр для каждой игры.

1. Starfield:
  - A. RPG,
  - B. roguelike,
  - C. научная фантастика,
  - D. открытый мир.
2. The Legend of Zelda: Tears of the Kingdom:
  - A. action-adventure,
  - B. гача,
  - C. открытый мир.
3. Hogwarts Legacy:
  - A. action RPG,
  - B. фэнтези,
  - C. открытый мир,
  - D. шутер.
4. Resident Evil 4 (Remake):
  - A. soulslike,
  - B. survival horror,
  - C. action-adventure.
5. Baldur's Gate 3:
  - A. RPG,
  - B. roguelite,
  - C. фэнтези,
  - D. пошаговая стратегия.

**Ответ:** BBDAB.

## Задача 2.4.2. Минималистичные циклы (6 баллов)

Темы: анализ игр, геймдизайн.

### Условие

Геймдизайнер хочет проверить, сможет ли его коллега лишь через прокачку персонажа, механику его смерти и механику битв составить такие игровые циклы, по которым можно точно определить похожие друг на друга игры.

Соотнесите игровые циклы, состоящие из ключевых пунктов: прокачки, битвы и гибели персонажа.

Таблица 2.4.1

	Рис. 2.4.1	Рис. 2.4.2	Рис. 2.4.3	Рис. 2.4.4	Рис. 2.4.5	Рис. 2.4.6
Darksouls						
Sifu						
CodeVein						
Hades 2						
Sekiro						
Metal gear rising						

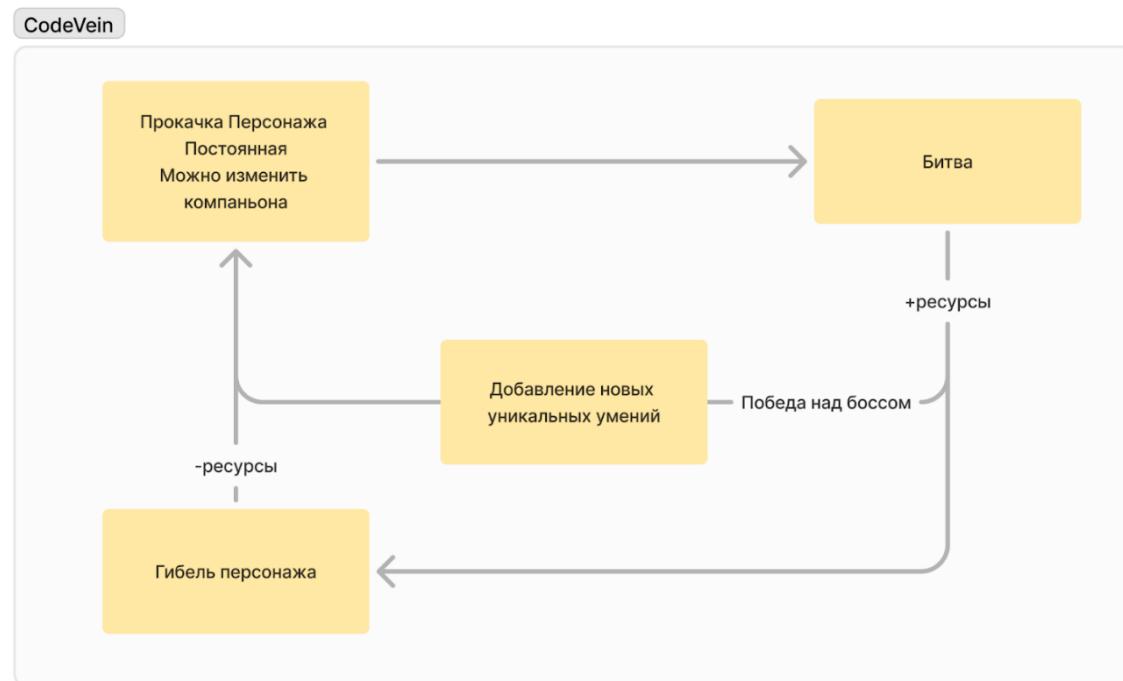


Рис. 2.4.1

## Dark souls

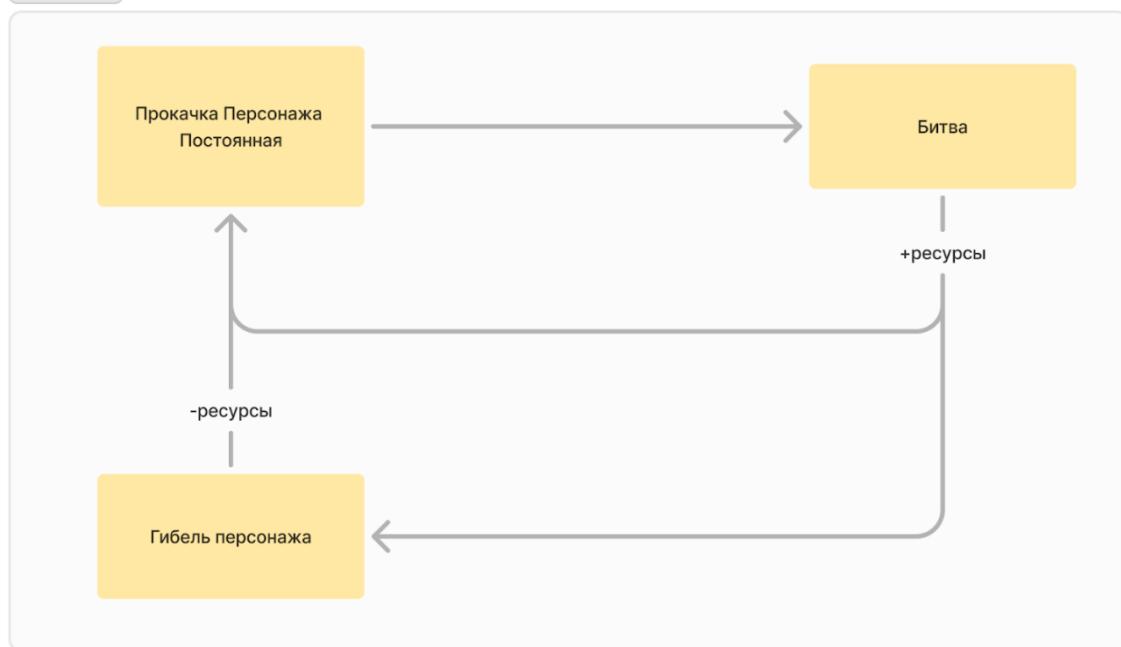


Рис. 2.4.2

## Metal gear rising

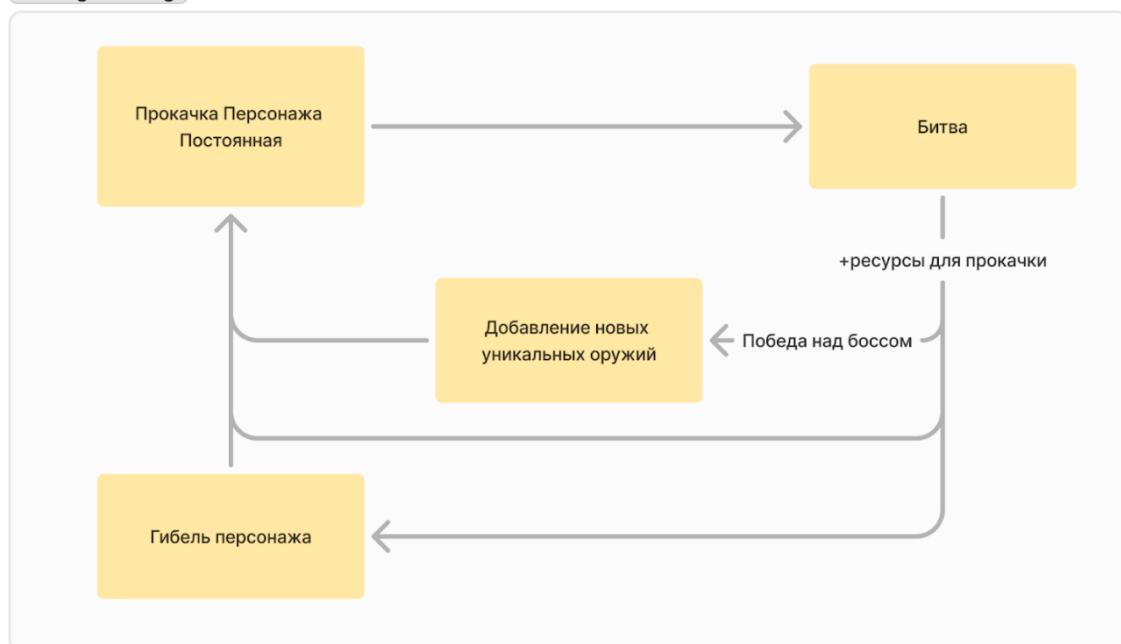


Рис. 2.4.3

Sekiro

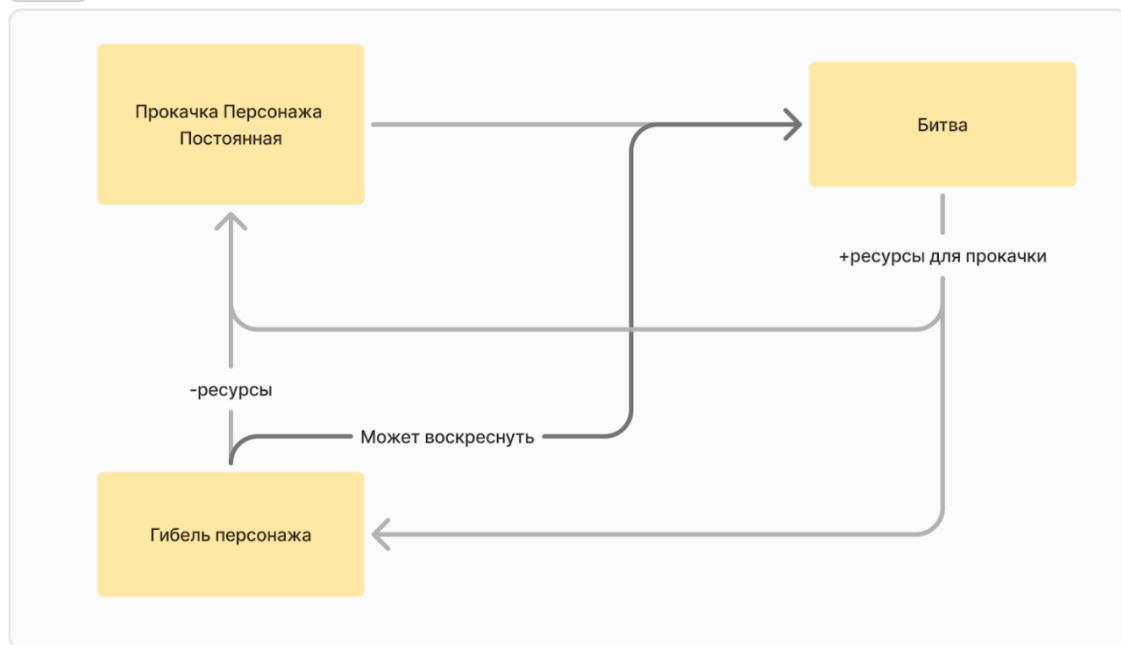


Рис. 2.4.4

Sifu



Рис. 2.4.5

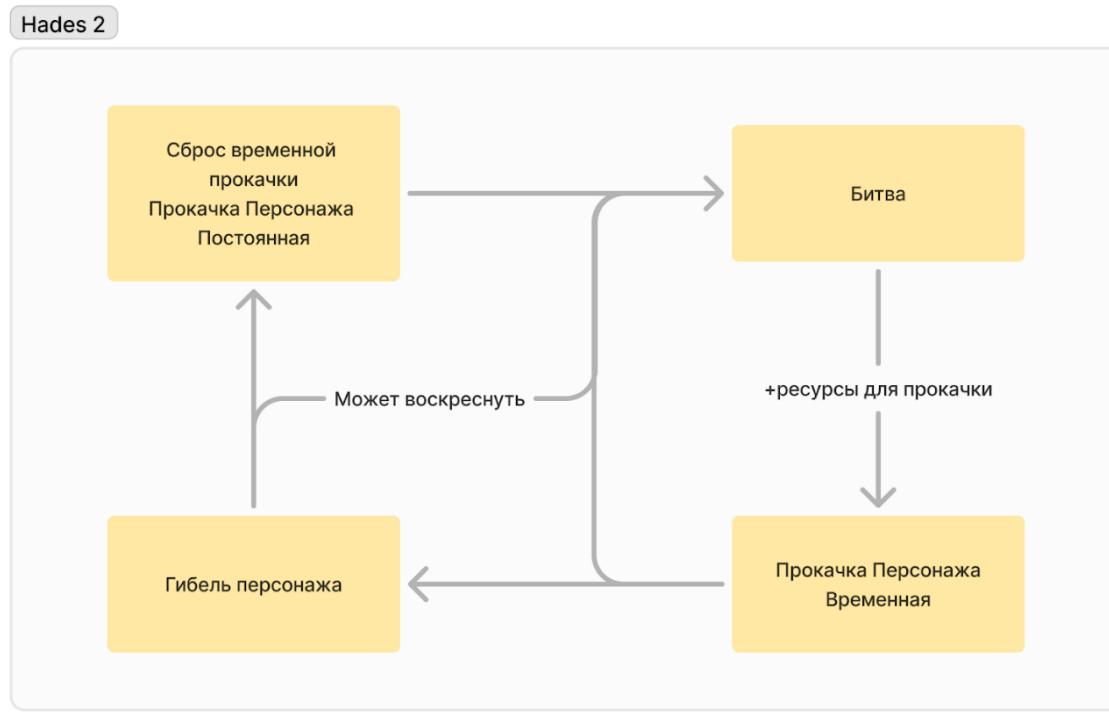


Рис. 2.4.6

**Ответ:** приведен в таблице [2.4.2](#).

Таблица 2.4.2

	Рис. 2.4.1	Рис. 2.4.2	Рис. 2.4.3	Рис. 2.4.4	Рис. 2.4.5	Рис. 2.4.6
DarkSouls						
Sifu						
CodeVein						
Hades 2						
Sekiro						
Metal gear rising						

### Задача 2.4.3. Качественный геймплей (6 баллов)

Тема: алгоритмы.

#### Условие

Инди-команде для конкурса пришла идея сделать простую игру-кликер, в которой с каждым тапом персонаж наносит урон врагу. Художник смог нарисовать шесть уникальных персонажей, для каждого из которых подготовил три разных окраски.

Геймдизайнер решил, что игрок будет сражаться сначала с первым уникальным персонажем с окраской 1, затем — с таким же, но окраской 2, потом — с окраской 3, а после этого будет сражаться со вторым уникальным персонажем с окраской 1 и так далее.

С каждой новой окраской персонаж становится сильнее персонажа с предыдущей окраской в три раза.

$(\text{Уникальный персонаж 1 Окраска 2}) = (\text{Уникальный персонаж 1 Окраска 1}) \cdot 3$ .

Каждый уникальный персонаж с окраской 1 равен силе предыдущего уникального персонажа с окраской 1 в 10 раз. То есть

$(\text{Уникальный персонаж 2 Окраска 1}) = (\text{Уникальный персонаж 1 Окраска 1}) \cdot 10$ .

Команде нужно определиться с начальным количеством силы для Уникального персонажа 1 с Окраской 1. Количество силы — это количество тапов, который должен сделать игрок, чтобы победить его и перейти к следующему противнику.

В ТЗ конкурса указано, что игрового времени нужно не менее 500 ч. При этом команда не хочет затягивать игру и стремится уложиться в минимальное время для прохождения.

Сколько нужно поставить силы Первому персонажу, если мы допускаем, что игрок делает три тапа за 1 с?

## Решение

Задачу можно выполнить, реализовав алгоритм решения на языке программирования Python.

Ниже представлено решение на языке Python.

### Python

```

1 personazi_kolvo = 6
2 colors = 3
3
4 def GetCurTaps(num, color, first_tap):
5     return first_tap * (10**num) * (3**(color))
6
7 def AllTaps(first_tap):
8     tapsCount = []
9     for i in range(personazi_kolvo):
10         for j in range(colors):
11             tapsCount += [GetCurTaps(i, j, first_tap)]
12     return tapsCount
13
14 # 500 hours = 500 * 3600 seconds = 1800000 seconds
15 player_tps = 3
16 border_time = 1800000
17
18 for nachTap in range(10):
19     sumTaps = sum(AllTaps(nachTap))
20     time = sumTaps / player_tps
21     if (time > border_time):
22         print(f"U must use minimum {nachTap} hp for first character")
23         break;

```

**Ответ:** 4 тапа.

### **Задача 2.4.4. Организация турнира (6 баллов)**

*Темы: комбинаторика, алгоритмы.*

#### **Условие**

Организаторы киберспортивного турнира хотят сэкономить и решают, какое минимальное число команд нужно взять на турнир, чтобы при этом количество матчей было не менее 90.

Правила турнира гласят: между любыми двумя командами должна быть сыграна одна игра. То есть каждая команда должна сыграть с каждой по одному разу.

Какое минимальное количество команд нужно взять на турнир, чтобы получилось не менее 90 игр?

#### **Решение**

Используя формулу комбинаторики, начинаем считать количество матчей с двух команд. Когда команд станет 14 — получится 91 матч.

$$14! / ((14 - 2)! \cdot 2!) = 14! / (12! \cdot 2!) = 13 \cdot 14 / 2 = 91 \text{ матч.}$$

**Ответ:** 14 команд.

### **Задача 2.4.5. Тест Unity (5 баллов)**

*Темы: программирование, инструменты разработки, Unity.*

#### **Условие**

Необходимо ответить на 10 вопросов, выбрав верный вариант ответа.

Ответом является числовое значение из пяти латинских букв без пробелов, например, ABBDC — значит, что ответ на первый вопрос был под номером А, ответ на второй вопрос под номером В, на третий — под номером В, на четвертый — под номером D, на пятый — под номером С.

1. За что отвечает профайлер?
  - A. Дает подробную информацию о папках и файлах проекта.
  - B. Показывает ошибки, возникающие в ходе работы.
  - C. Оценивает производительность приложения.
  - D. Является настройкой, применимой к объекту.
2. Как следует настроить луч, если мы хотим проверить его на пересечение с заданным объектом?
  - A. Проверка всех объектов на пути луча по `layer`.
  - B. Проверка всех объектов на пути луча по имени.

- C. Отключение коллайдеров у объектов, которые могут встретиться на пути.
- D. Расчет угла от отрезка «объект-цель» и направления взгляда.
3. В какой папке необходимо содержать скрипт, отвечающий за отображение свойств другого скрипта в редакторе?
- A. Любая папка `Editor` проекта.
  - B. Не нужна специальная папка.
  - C. Папка `Editor` в папке выше по иерархии, чем в которой находится целевой скрипт.
  - D. Папка `Editor` в той же папке, что и целевой скрипт.
4. Какой из методов сработает, когда объект будет внутри какого-то триггера?
- A. `OnTriggerEnter2D`.
  - B. `OnCollisionEnter2D`.
  - C. `OnCollisionExit2D`.
  - D. `OnTriggerStay2D`.
5. Зачем нужен прием LOD?
- A. Позволяет снизить нагрузку на графический вычислитель.
  - B. Позволяет тратить меньше времени разработчика на моделирование объектов.
  - C. Позволяет упростить взаимодействие с `Input`-системой.
  - D. Позволяет сократить использование ресурсов компьютера за счет рендеринга исключительно видимых объектов.

**Ответ:** CADDA.

### **Задача 2.4.6. Плохая связь (15 баллов)**

*Тема: программирование.*

#### **Условие**

Новый мэр поселка городского типа в своей предвыборной программе обещал исправить печальную ситуацию со связью.

В городе есть  $N$  важных объектов, которым требуется нормальная связь. Цель — обеспечить связью как минимум  $K$  из них. Место размещения сотовой вышки имеет координаты  $[X_0; Y_0]$ .

Для вышки связи определяется радиус покрытия  $R \geq 0$ . Все здания на расстоянии не более  $R$  от вышки считаются обеспеченными связью. Здания на расстоянии больше  $R$  от вышки уже не входят в зону гарантированного покрытия, и связь работает с перебоями. Пусть  $D$  — расстояние от вышки до сооружения. Вероятность работы связи  $P(D, R)$  задана такой формулой:

$$P(D, R) = \begin{cases} 1, & : D \leq R, \\ \exp\left(1 - \frac{D^2}{R^2}\right), & : D > R. \end{cases}$$

Под  $\exp(a)$  следует понимать  $e^a$ , где  $e = 2,7182818284590452353602874713527$ .

Если радиус покрытия равен нулю, то все объекты, находящиеся в точке размещения вышки, будут обеспечены связью, а все остальные — нет.

Бюджет на оборудование крайне ограничен, отчего начальство просит рассчитать минимальный радиус покрытия для вышки, который позволит обеспечить связью объекты с суммарной вероятностью НЕпокрытия для всех объектов не более, чем  $\varepsilon$ .

### ***Формат входных данных***

В первой строке записано целое число  $N$  — количество целевых объектов ( $1 \leq N \leq 100$ ). Во второй строке записано два целых числа:  $K$  — целевое количество обеспеченных связью объектов, и  $\varepsilon$  — максимально допустимая вероятность не обеспечить связь, заданная в промилле ( $1 \leq K \leq N$ ,  $1 \leq \varepsilon \leq 999$ , в 1 единице 1 000 промилле). В третьей строке записаны  $X_0$  и  $Y_0$  — координаты точки размещения вышки. Остальные  $N$  строк содержат по два числа  $X_i$  и  $Y_i$  — координаты каждого объекта. Все координаты целые, по модулю не превосходят 1 000. В одной точке может находиться несколько объектов.

### ***Формат выходных данных***

Выведите искомый расчетный радиус покрытия для сотовой вышки. В ответе необходимо указать положительное число с точностью до тысячных долей, по правилу округления все остальные знаки отбрасываются, последняя значащая цифра не изменяется.

### ***Решение, тесты и ответ***

<https://disk.yandex.ru/d/rThzp5cSSJ1JPg>.

### ***Задача 2.4.7. Вспоминая прошлое (20 баллов)***

*Тема: программирование.*

Особенности начисления баллов: начисление баллов пропорционально пройденным тестам.

### ***Условие***

Программист Вася захотел написать свою собственную игру и решил начать с простого. Используя написанный Васей код, нужно реализовать функцию `MoveHero` в классе `Field`.

Правила игры:

1. Основные элементы (классы) игры:

- A. `Field` — прямоугольное игровое поле. Содержит переменные, отвечающие за ширину и высоту поля. Обе переменные имеют целочисленное четное значение в диапазоне (10, 100).

- B. Obstacle — прямоугольное препятствие на поле. Содержит переменные, отвечающие за ширину и высоту препятствия. Обе переменные имеют целочисленное четное значение в диапазоне (2, 20), а также переменные, отвечающие за координаты на поле. За координаты объекта принимается его геометрический центр.
- C. Hero — квадратный главный герой. Содержит переменные, отвечающие за ширину и высоту героя. Обе переменные имеют целочисленное четное значение в диапазоне (2, 6). Также имеет переменные, отвечающие за координаты на поле. За координаты объекта принимается его геометрический центр.
- D. DirectionVector — вектор перемещения. Отвечает за смещение персонажа по полю. Содержит два целочисленных значения в диапазоне (-1, 1).

2. Правила движения персонажа по полю:

- A. Поле представляет собой координатную сетку, где левый нижний угол имеет координаты (0, 0), а правый верхний (width, height).
- B. Персонаж может перемещаться по полю на одну клетку по вертикали, по горизонтали и по диагонали.
- C. Если координата персонажа пересекает в процессе хода границу поля, то он переносится на другую сторону поля. Допустим, поле имеет размер 50 единиц в ширину и 20 — в высоту. Тогда координаты игрока будут изменяться при движении по горизонтали так: (49, 10) → (50, 10) → → (1, 10) → (2, 10), аналогично по вертикали и в обратную сторону. В случае движения по диагонали: (49, 19) → (50, 20) → (1, 1) → (2, 2). Обратите внимание: в этом правиле не учитывается размер персонажа, исключительно — его координаты. Если персонаж на своем текущем ходу пересек поле (например, (50, 10) → (1, 10)), необходимо вывести в консоль `Teleported`.
- D. Персонаж в начале игры расположен в центре поля.
- E. На поле расположены прямоугольные препятствия. По дизайну игры персонаж не может заходить никакой своей частью на препятствие (возникает столкновение). В случае, если при попытке движения персонажа он сталкивается с препятствием, необходимо вывести в консоль (в зависимости от направления столкновения) либо `We got collision by Y`, либо `We got collision by X`, либо `We got collision by X and Y`, если персонаж столкнулся с препятствием по оси *Y*, по оси *X* или с углом препятствия соответственно. Обратите внимание: если персонаж пытается переместиться по обеим осям одновременно, перемещение будет заблокировано (и не должно произойти) только по той/тем оси/осям, перемещение по которым невозможно из-за столкновения (если персонаж столкнулся с препятствием сверху, то при попытке перемещения влево-вверх персонаж переместится влево). В столкновении учитываются границы (ширина и высота) персонажа и препятствия.  
Кроме того, если препятствие не мешает текущему ходу персонажа, не надо выводить сообщение о столкновении.
- F. Гарантируется, что персонаж не будет начинать игру и при этом находится любой своей частью на части препятствия.

- G. Гарантируется, что заданные во входных данных препятствия полностью находятся на поле и не пересекают его границ.
- H. Гарантируется, что заданные во входных данных препятствия не создадут ситуации, при которой игрок может столкнуться с двумя препятствиями одновременно.
- I. Гарантируется, что заданные во входных данных препятствия не создадут ситуации, что при «перепрыгивании» персонажа через границу экрана он попадет какой-либо частью своей «модели» на «модель» препятствия.

### ***Формат входных данных***

В первой строке записано целое число  $N$  — количество препятствий ( $1 \leq N \leq 100$ ). Во второй строке записано два целых числа: ширина `width` и высота `height` персонажа. В третьей строке записано два целых числа: ширина `width` и высота `height` игрового поля. Следующие  $N$  строк содержат по четыре числа: ширина `width` и высота `height` препятствия, а также  $X_i$  и  $Y_i$  — его координаты. Все оставшиеся строки содержат пару чисел  $X_{dir}$  и  $Y_{dir}$ , отвечающих за вектор перемещения персонажа согласно описанию.

Обратите внимание: все входные данные обрабатываются программой тестирования, которая затем вызывает функцию `MoveHero` каждый ход и передает в нее корректный `DirectionVector`. Ввод данных обрабатывать не нужно.

### ***Формат выходных данных***

Перед первым ходом программа выводит текущие координаты игрока в формате  $(X, Y)$ . Далее каждый ход программа вызывает функцию `MoveHero` (в которой необходимо реализовать вывод указанных выше сообщений), после чего выводит координаты игрока после хода в формате  $(X, Y)$ .

Обратите внимание: необходимо выводить только указанные диагностические сообщения в случае пересечения экрана и столкновения с препятствием. Координаты игрока программа тестирования выводит автоматически.

### ***Решение, тесты и ответ***

<https://disk.yandex.ru/d/h60ZRw2Z-PI5-Q>.

### ***Задача 2.4.8. 3D тест (4 балла)***

Темы: 3D-моделирование, 2D-арм.

### ***Условие***

Необходимо ответить на восемь вопросов, выбрав верный вариант ответа.

Ответом является числовое значение из пяти латинских букв без пробелов, например, ABBDC — значит, что ответ на первый вопрос был под номером А, ответ

на второй вопрос под номером В, на третий — под номером В, на четвертый — под номером D, на пятый — под номером С.

1. Что такое UI в разработке игры?
  - A. Пользовательский интерфейс.
  - B. Пользовательский ввод.
  - C. Унифицированный интерфейс.
  - D. Универсальный интерфейс.
2. Что такое концепт-арт?
  - A. Исходный материал для игры.
  - B. Предварительный дизайн персонажей и локаций.
  - C. Конечный спрайт игрока.
  - D. Стиль рисования.
3. Какие инструменты можно использовать для создания 3D-моделей?
  - A. Blender.
  - B. ZBrush.
  - C. SketchUp.
  - D. Все вышеперечисленное.
4. Какие методы использования света могут делать игровую графику более привлекательной?
  - A. Освещение в реальном времени.
  - B. Глубокий тени.
  - C. Туман и рассеянный свет.
  - D. Все вышеперечисленное.
5. Какие типы картин могут относиться к 2D-арту?
  - A. Фотографии.
  - B. Иллюстрации.
  - C. Комиксы.
  - D. Все варианты выше.

**Ответ:** ABDDD.

### **Задача 2.4.9. Черный ящик (5 баллов)**

*Темы: 3D, моделирование.*

#### **Условие**

Скачайте архив с 3D-моделью: [https://disk.yandex.ru/d/\\_avd\\_uKNlsWVmQ](https://disk.yandex.ru/d/_avd_uKNlsWVmQ).

Определите количество треугольников (**Tris**) у модели, которая лежит внутри ящика.

Модель необходимо скачать по ссылке из текстового файла задания. В поле ответа впишите количество треугольников.

Рекомендуем пользоваться самой свежей версией программы **Blender**.

### ***Решение***

Определить количество треугольников можно с помощью любого ПО, позволяющего работать с 3D-графикой, используя либо встроенные модификаторы/плагины, либо в режиме редактирования моделей. Но для начала необходимо удалить ящик, который закрывает обзор фигурки: необходимо перейти в режим редактирования и выделить все вершины ящика, затем удалить их. После этого можно узнать количество треугольников у фигурки.

**Ответ:** 19 600.

### ***Задача 2.4.10. Перепутанные крышки (6 баллов)***

**Темы:** 3D-моделирование, текстурные карты.

#### ***Условие***

Скачайте архив с текстурными картами и 3D-моделью: [https://disk.yandex.ru/d/8mBrxrw\\_e5XpxA](https://disk.yandex.ru/d/8mBrxrw_e5XpxA).

Наложите текстурные карты на предложенные модели, в соответствии с номерами каждого объекта (то есть фигуре «1» принадлежат текстуры с именем «1»). Можно заметить, что некоторые крышки перепутаны. Передвиньте крышки на свои места (ориентируясь на цвет флакона) и впишите в поле ответа в правильном порядке цифры, изображенные на крышке. Рекомендуем пользоваться самой свежей версией программы **Blender**.

### ***Решение***

Необходимо скачать архив с текстурными картами и 3D-моделью. Открыть модель в **Blender** и импортировать ее. После загрузить текстурные карты и применить их к соответствующим объектам на модели, для этого необходимо создать материалы и назначить текстуры. Также необходимо проверить соответствие текстурных карт номерам каждого объекта. Если обнаружены ошибки (когда крышки перепутаны), переместить крышки на правильные места, ориентируясь на цвет флакона. После перестановки крышечек нужно определить цифры, изображенные на крышках, в правильном порядке и вписать их в поле ответа.

**Ответ:** 2406.

### ***Задача 2.4.11. Сломанная гитара (7 баллов)***

**Темы:** 3D-моделирование, анимация, ключи.

#### ***Условие***

Скачайте архив с 3D-моделью: <https://disk.yandex.ru/d/JmyNR60CquwUbA>.

В файле находится гитара и три ноты. Следует внимательно просмотреть анимацию и найти проблему, которая мешает ноте № 3 совершить оборот вокруг гитары. Исправьте этот фрагмент, чтобы нота продолжила маршрут. После обнаружения и исправления проблемы, необходимо посчитать количество оборотов ноты № 3 вокруг гитары (начало круга идет от струн гитары).

Рекомендуем пользоваться самой свежей версией программы **Blender**.

### **Решение**

Скачайте архив с 3D-моделью гитары и нотами. Откройте файл в **Blender** и просмотрите анимацию, чтобы найти проблему, которая мешает ноте № 3 совершить оборот вокруг гитары. Проблема связана с неправильной анимацией ключевых кадров. Необходимо найти фрагмент и удалить данные ключевые кадры, чтобы нота № 3 могла успешно продолжить маршрут вокруг гитары. После того как проблема исправлена, посчитайте количество оборотов ноты № 3 вокруг гитары. Запишите ответ.

**Ответ:** 3.

### **Задача 2.4.12. Воронка (5 баллов)**

Темы: маркетинг, проценты.

### **Условие**

Необходимо проанализировать воронку продаж игры и определить упущеный процент конверсии. Затем следует найти этап, на котором процент конверсии значительно ниже, чем мог быть, найдите его.

Запишите процент и количество людей на этапе, в котором самая высокая конверсия.

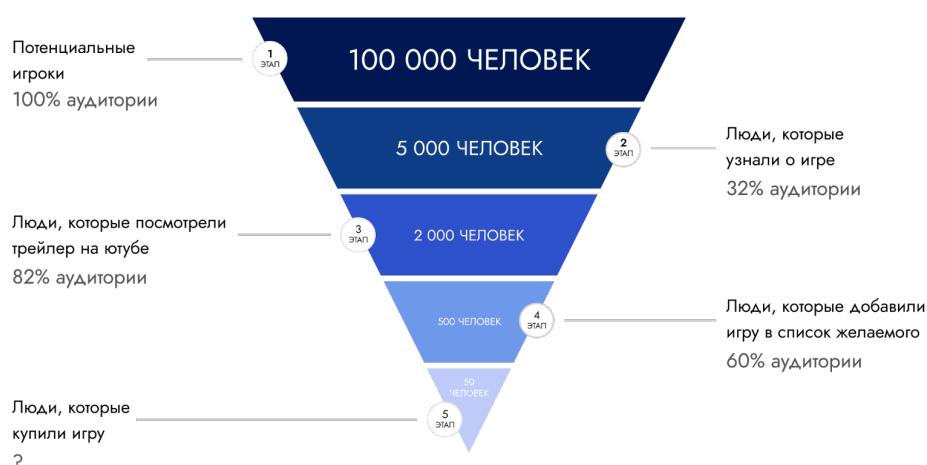


Рис. 2.4.7

### ***Решение***

Необходимо посчитать количество людей из категории «Люди, которые добавили игру в список желаемого»:

$$26240/100 \cdot 60 = 15744.$$

Необходимо посчитать процент из категории «Люди которые купили игру»:

$$1103/15744 \cdot 100 = 7.$$

Посмотрев на воронку (рис. 2.4.7), можно увидеть, что самая высокая конверсия у категории «Люди, которые посмотрели трейлер на YouTube».

Потенциальные игроки — 100% — 100 000.

Люди, которые узнали об игре — 32% — 32 000.

Люди, которые посмотрели трейлер на YouTube — 82% — 26 240.

Люди, которые добавили игру в список желаемого — 60% — 15 744.

Люди, которые купили игру — 7% — 1 103.

**Ответ:** 82, 26 240.

### ***Задача 2.4.13. Сложные вопросы (5 баллов)***

Темы: геймдизайн, теория вероятности.

#### ***Условие***

В новеллу «Журнал капитана Тундры» (<https://talent.kruzhok.org/novel/>) включены вопросы с образовательным содержанием, предполагающие, что игрок при выборе ответа использует знания из школьной программы.

Определите, пытаются ли игроки ответить на эти вопросы правильно или все просто кликают в случайный ответ. Для этого нужно сравнить реальную долю ответивших верно на все вопросы с долей, которая должна была сделать это, если бы все игроки отвечали случайным образом. Для исследования были взяты сцены в аэротакси и при разговоре с ИИ. Рассчитайте, какая доля игроков в процентах должна была бы ответить верно на все вопросы с образовательным содержанием в первом эпизоде новеллы с округлением до десятых, если бы ответы давались случайно.

### ***Решение***

В первом эпизоде новеллы в сценах в аэротакси и при разговоре с ИИ всего два вопроса отсылающих к школьной программе: про ускорение и тягу двигателя. На каждый из них предлагается по три варианта ответа, где только один верный. Ответы на вопросы даются независимо, так что эти события совместные. Вероятность произведения нескольких совместных событий равна произведению вероятностей этих событий. Вероятность случайного верного ответа будет  $1/3$  для каждого вопроса, то есть  $1/9$  для обоих вопросов. В переводе в проценты из попыток это даст 11 и 1 в периоде процентов, то есть 11,1%.

**Ответ:** 11,1%.

### **Задача 2.4.14. Определение целевой аудитории (5 баллов)**

Темы: геймдизайн, маркетинг игры.

#### **Условие**

При подготовке презентации для рекламного агентства с целью предложить интеграцию в визуальную новеллу «Журнал капитана тундра» (<https://talent.kruzhok.org/novel/>) было запланировано включение в нее слайда о целевой аудитории игры. Выберите, какие тезисы будет корректно использовать в такой презентации, чтобы описать составляющие игры и аудиторию, на которую они рассчитаны.

Ответ давайте на основе выводов, которые можно сделать из первого эпизода новеллы.

1. Аниме-стиль визуальной новеллы будет привлекателен для широкой аудитории.
2. Упрощенная, яркая графика персонажей привлечет детей дошкольного возраста.
3. Сюжет о сложных отношениях с родителями и построении коммуникации в команде сверстников будет актуален для подростков.
4. Сюжет новеллы строится вокруг научного исследования, поэтому будет интересен ученым.
5. Новелла привлекает игроков, которым нравятся игры с интересными сюжетом.
6. Подросткам будет интересно играть в новеллу с персонажами-сверстниками.

#### **Решение**

Данная новелла не сделана в стиле аниме, поэтому это неверно, кроме того, утверждение про широкую аудитории не несет ценности при определении целевой аудитории.

Графика основных персонажей действительно упрощенная и могла бы быть интересна дошкольникам, но поскольку новелла предполагает большой объем текста, это будет также некорректно для презентации о возможных рекламных интеграциях.

Сюжет новеллы в первом эпизоде построен в основном вокруг отношений героя-подростка со сверстниками, наставницей и отцом, что, судя по популярности литературы о подростках, действительно может быть интересно читателям того же возраста.

В целом новелла как жанр привлекает тех, кому интересен сюжет, так что может быть интересна для рекламы других аналогичных игр.

**Ответ:** 1. — неверно; 2. — неверно; 3. — верно; 4. — неверно; 5. — верно; 6. — верно.

### **3. Второй отборочный этап**

#### **3.1. Работа наставника НТО на этапе**

На втором отборочном этапе НТО участникам предстоит решать как индивидуальные, так и командные задачи в рамках выбранного профиля. Подготовка к этому этапу требует от них не только глубокого понимания предметной области, но и умения работать в команде, эффективно распределять роли и применять полученные знания на практике. Наставник играет здесь важную роль — он помогает участникам выстроить осмысленную и целенаправленную траекторию подготовки.

Вот основные направления, в которых наставник может поддержать участника:

- **Подготовка по образовательным программам НТО.** Наставник может готовить участников, используя готовые образовательные программы по технологическим направлениям, рекомендованные организаторами, а также адаптировать их под уровень подготовки школьников.
- **Разбор заданий прошлых лет.** Изучение задач второго отборочного этапа прошлых лет помогает участникам понять формат заданий, определить типовые ошибки и выработать стратегии решения.
- **Онлайн-курсы.** Участники могут пройти курсы по разбору задач прошлых лет или курсы, рекомендованные разработчиками отдельных профилей. Наставник может включить эти курсы в план подготовки, а также сопровождать процесс изучения и помогать с возникшими вопросами.
- **Анализ материалов профиля.** Совместный разбор методических материалов, размещенных на страницах профилей, помогает уточнить требования к участникам и направить подготовку на ключевые темы.
- **Практикумы.** Это важный элемент подготовки, позволяющий применять знания на практике. Наставник может:
  - ◊ организовать практикумы по методическим материалам с сайта профиля;
  - ◊ декомпозировать задачи заключительного этапа прошлых лет на отдельные элементы и проработать их с участниками;
  - ◊ провести анализ требуемых профессиональных компетенций и спланировать занятия для развития наиболее значимых из них;
  - ◊ направить участников на практикумы и мероприятия от организаторов, которые анонсируются в официальных сообществах НТО, например, в телеграм-канале для наставников: [https://t.me/kruzhok\\_association](https://t.me/kruzhok_association).
- **Командная работа.** Одной из ключевых задач наставника на втором этапе является помочь в формировании команды или в поиске подходящей. Наставник может помочь участникам определить их сильные стороны, выбрать роль в команде и сориентироваться в процессе командообразования, включая участие в бирже команд в рамках конкретного профиля.

## *Если участники не прошли отборочный этап*

Случается, что несмотря на усилия и серьезную подготовку, участники не проходят во второй или заключительный этап Олимпиады. В такой ситуации особенно важна поддержка наставника.

- **Поддержка и признание усилий.** Наставнику важно подчеркнуть ценность пройденного пути: полученные знания, навыки, преодоленные трудности и личностный рост. Это помогает участникам сохранить мотивацию и не воспринимать результат как окончательное поражение.
- **Рефлексия.** Полезно организовать встречу для обсуждения впечатления от участия, трудности, с которыми столкнулись школьники и то, что они узнали о себе и команде. Наставник может направить разговор в конструктивное русло: какие выводы можно сделать? Что сработало хорошо? Что можно улучшить?
- **Анализ ошибок и пробелов.** Наставник вместе с участниками анализирует, какие темы вызвали наибольшие затруднения, чего не хватило в подготовке — теоретических знаний, практических навыков, командного взаимодействия. Это позволяет выстроить более эффективную стратегию на будущее.
- **Планирование дальнейшего пути.** Участникам можно предложить:
  - ◊ продолжить углубленное изучение профиля или смежных направлений;
  - ◊ заняться проектной деятельностью, которая укрепит знания и навыки;
  - ◊ сформировать план по подготовке к следующему циклу НТО, начиная с работы над типовыми заданиями и курсами.
- **Создание устойчивой мотивации.** Важно показать школьникам, что участие в НТО — это не просто соревнование, а часть большого образовательного маршрута. Даже неудачный результат может стать толчком к профессиональному росту, если воспринимать его как точку развития, а не как конец пути.

Таким образом, наставник помогает участникам не только готовиться к этапам НТО, но и справляться с неудачами, выстраивать долгосрочную стратегию и сохранять интерес к инженерному и технологическому творчеству.

## 3.2. Инженерный тур

### 3.2.1. Индивидуальные задачи

Индивидуальные задачи второго этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Конкурс: <https://contest.yandex.ru/contest/69902/enter/>.

#### Геймдизайн

##### Задача 3.2.1.1. Презентация жанров (10 баллов)

Темы: геймдизайн, жанры, нарратив.

#### Условие

Ленивый Геймдизайнер из задания первого этапа наконец-то наигрался в разные игры. Даже собрал команду из одноклассников.

После долгих размышлений он остановился на шести жанрах, но понял, что команда не так хорошо разбирается в них (для них шутеры — это стрелялки, ужас какой!). Поэтому попросил свою любимую нейросеть сделать описания жанрам и прикрепить скриншот из игры.

Он захотел поэкспериментировать и проверить насмотренность сокомандников. Распечатал все на отдельных карточках. Но когда споткнулся, почти все карточки разлетелись. Помогите Ленивому Геймдизайнеру составить таблицу жанров правильно.

Таблица 3.2.1

№	Жанр	Аспекты жанра	Скриншот из игры
1	Метроидвания		
2	Экшен-приключение		
3			

№	Жанр	Аспекты жанра	Скриншот из игры
4		Игровой процесс состоит из секций. В каждой секции имеется проблема, которую нужно решить, чтобы продвинуться дальше. Игрок может быть как абстракцией, так и иметь конкретную личность в рамках мира. Игрока редко ограничивают во времени прохождения секций.	
5		Игровой процесс состоит из отрезков, в рамках которых предпринимаются какие-то действия. Игрок может представлять как одного героя, так и целый отряд. Игрок может достаточно долго раздумывать перед совершением действий.	
6			

### Задание

Заполните таблицу.

1. Расположите следующие **жанры** в таблице правильном порядке:
  - A. Головоломка.
  - B. Пошаговая тактика.
  - C. Симулятор.
  - D. Пошаговая стратегия.
2. Расположите следующие **аспекты жанров** в таблице правильном порядке:
  - A. Игровой процесс состоит из множества локаций, которые связаны прогрессией сложности и возможностей. Игрок может быть чем угодно, но редко представляет собой абстракцию. Ограничений по времени нет, сложность заключается в узких таймингах и правильном порядке применения возможностей.
  - B. Игровой процесс состоит из порядка действий, которые подражают действительности в нескольких аспектах. Зачастую игрок — представитель какой-то профессии. Часто нет ограничений по времени, но в некоторых случаях за высокую скорость можно получить дополнительную награду.
  - C. Игровой процесс состоит из отрезков, в рамках которых предпринимаются различные действия, одному из поджанров характерна модель из 4 таких действий. Игрок зачастую представляет государства, но некоторые игры берут еще больший масштаб. Он может достаточно долго раздумывать над совершением действий.
  - D. Игровой процесс включает множество различных сцен, наполненных событиями. Есть главная механика, но вдобавок к ней разработчики для разнообразия создают множество гиммиков для привлечения и удержания аудитории. Почти всегда игрок — проработанный персонаж, органично вписанный в мир. Чаще всего действия происходят в реальном

времени. В рамках цели может ставиться ограничение по времени, чтобы заставить игрока действовать быстро.

3. Расположите скриншоты из игр в таблице правильном порядке:

- A. <https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P1.jpg>;
- B. <https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P2.jpg>;
- C. <https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P3.jpg>;
- D. <https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P4.jpg>.

### **Решение**

Таблица 3.2.2

<b>№</b>	<b>Жанр</b>	<b>Аспекты жанра</b>	<b>Скриншот из игры</b>
1	Метроидвания	Игровой процесс состоит из множества локаций, которые связаны прогрессией сложности и возможностей. Игрок может быть чем угодно, но редко представляет собой абстракцию. Ограничений по времени нет, сложность заключается в узких таймингах и правильном порядке применения возможностей.	<a href="https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P4.jpg">https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P4.jpg</a>
2	Экшен-приключение	Игровой процесс включает множество различных сцен, наполненных событиями. Есть главная механика, но вдобавок к ней разработчики для разнообразия создают множество гиммиков, для привлечения и удержания аудитории. Почти всегда игрок — проработанный персонаж, органично вписанный в мир. Чаще всего действия происходят в реальном времени. В рамках цели может ставиться ограничение по времени, чтобы заставить игрока действовать быстро.	<a href="https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P2.jpg">https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P2.jpg</a>
3	Симулятор	Игровой процесс состоит из порядка действий, которые подражают действительности в нескольких аспектах. Зачастую игрок — представитель какой-то профессии. Часто нет ограничений по времени, но в некоторых случаях за высокую скорость можно получить дополнительную награду.	
4	Головоломка	Игровой процесс состоит из секций. В каждой секции имеется проблема, которую нужно решить, чтобы продвинуться дальше. Игрок может быть как абстракцией, так и иметь конкретную личность в рамках мира. Его редко ограничивают во времени прохождения секций.	<a href="https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P3.jpg">https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P3.jpg</a>
5	Пошаговая тактика	Игровой процесс состоит из отрезков, в рамках которых предпринимаются какие-то действия. Игрок может представлять как одного героя, так и целый отряд. Он может достаточно долго раздумывать перед совершением действий.	<a href="https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P1.jpg">https://disk.yandex.ru/d/oTFH6tlWuo4aAw/P1.jpg</a>

Таблица 3.2.2

№	Жанр	Аспекты жанра	Скриншот из игры
6	Пошаговая стратегия	Игровой процесс состоит из отрезков, в рамках которых предпринимаются различные действия, одному из поджанров характерна модель из 4 таких действий. Он зачастую представляет государства, но некоторые игры берут еще больший масштаб. Игрок может достаточно долго раздумывать над совершением действий.	

### Задача 3.2.1.2. Тяжелый мед (10 баллов)

Темы: геймдизайн, игровые механики.

#### Условие

Жанр выбран и даже написан сюжет! Программист Матвей пишет механики по техническому заданию, но напутал значения. В итоге получается либо имбаланс, либо что-то бесполезное. Геймдизайнер врывается и требует тетрадку (балансную таблицу) у Матвея, но оказывается, ее погрыз алабай Матвея.

Чтобы восстановить таблицу, Геймдизайнер нашел голосовое сообщение, которое давно отправлял маме, где описывал свою логику баланса. Расшифровка голосового сообщения:

«Привет, мам, я такую игру классную придумал! Нет, это не как в тот раз, грабить караваны там не надо, она полезная даже. Я еще механику такую классную придумал, сейчас расскажу, я ее назвал «Сюрприз-механика», никогда такую не встречал. И нет, мам, это не лутбоксы!!!

Ближе к механике. Главный герой может ходить работать в Академию, после смены он получает случайный предмет, который имеет разную ценность в зависимости от редкости. Самое редкое, что можно незаметно вынести — это артефакт древних медведей. В 5 раз чаще артефакта выпадает прототип дрона-разведчика. В 3 раза реже прототипа выпадает лазерный резак, а в 12 раз чаще — бочонок энергомеда. Еще есть кристалл памяти, он в 13 раз чаще выпадает, чем набор для хакинга. А набор для хакинга в 4 раза более редкий, чем бочонок. Фух, как скороговорка какая-то, ну как тебе?»

По заданию Олимпиады команде геймдизайнера нужно добавить валюту в их игру, и чтобы каждый предмет имел стоимость. Программист предложил стоимость предметов считать обратно пропорциональной вероятности получения предмета после смены.

Сколько понадобится условной валюты, чтобы купить: один набор хакера, пять бочонков меда, два лазерных резака суммарно (цена каждой единицы предмета округляется вверх)?

### **Решение**

Поскольку все вероятности можно представить как  $P_{\min} \cdot X$ , то будем записывать именно множители. Получаем следующую таблицу.

Таблица 3.2.3

Кристалл памяти	Артефакт древних медведей	Улучшенный лазерный резак	Набор для хакера	Прототип дрона-разведчика	Бочонок меда	
195	1	1,666666667	15	5	60	

Чтобы избавиться от дробей, все умножаем на 3 и понимаем, что сумма вероятностей должна равняться единице. Поэтому рассчитываем сумму этих элементов.

Таблица 3.2.4

Кристалл памяти	Артефакт древних медведей	Улучшенный лазерный резак	Набор для хакера	Прототип дрона-разведчика	Бочонок меда	
195	1	1,666666667	15	5	60	
585	3	5	45	15	180	833

Делим каждое значение на сумму значений, чтобы получить долю (вероятность) получить тот или иной предмет. По определению из задания считаем стоимость (обратное от вероятности).

Таблица 3.2.5

Кристалл памяти	Артефакт древних медведей	Улучшенный лазерный резак	Набор для хакера	Прототип дрона-разведчика	Бочонок меда	
195	1	1,666666667	15	5	60	
585	3	5	45	15	180	833
0,702280912	0,003601441	0,006002401	0,054011609	0,018007203	0,216064545	
1,423931624	277,666667	166,6	18,51111111	55,53333333	4,62777778	

Округляем нужное, умножаем на количество и суммируем.

Таблица 3.2.6

Кристалл памяти	Артефакт древних медведей	Улучшенный лазерный резак	Набор для хакера	Прототип дрона-разведчика	Бочонок меда	
195	1	1,666666667	15	5	60	
585	3	5	45	15	180	833
0,702280912	0,003601441	0,006002401	0,054011609	0,018007203	0,216064545	
1,423931624	277,666667	166,6	18,51111111	55,53333333	4,62777778	
		167	19		5	

Кристалл памяти	Артефакт древних медведей	Улучшенный лазерный резак	Набор для хакера	Прототип дрона-разведчика	Бочонок меда	
		334	19		25	378

**Ответ:** 378.

### **Задача 3.2.1.3. Пасхантеры, соберитесь! (15 баллов)**

**Тема:** геймдизайн.

#### **Условие**

Ленивый Геймдизайнер параллельно с Олимпиадой пробился в реальную игровую индустрию и помогал студии Ninsar.games в разработке игры «Сатурн». Он очень любит пасхалки, поэтому решил оставить одну на свою любимую Олимпиаду, но чтобы не выбиваться из сеттинга, решил делать отсылку не на свой профиль.

Время шло, Геймдизайнер решил похвастаться команде, но забыл профиль Олимпиады, на который сделал отсылку. Он помнит лишь то, что оставил ее в спальне Акима, где-то около кровати.

Скачайте игру в VK Play, либо на официальном сайте, либо в Steam и найдите отсылку на один из профилей НТО.

Ответом является полное название этого профиля. Например: Разработка компьютерных игр.

VK Play: <https://vkplay.ru/play/game/saturn/>.

Официальный сайт: <https://ninsar.games/saturn/ru>.

Steam: <https://store.steampowered.com/app/2742840/Saturn/?l=russian>.

#### **Решение**

1. Скачать игру.
2. Подняться из подвала Акима на второй этаж его дома.
3. Найти объект.
4. Прочесть Записку.
5. Написать ответ.

**Ответ:** Технологии беспроводной связи.

#### **Арт**

Ответы на все три задания по компетенции Арт кроются в файле по ссылке: <https://disk.yandex.ru/d/fn0-KCOS2N81aQ>.

### **Задача 3.2.1.4. Вертексы и треугольники (10 баллов)**

*Темы: арт, 3D-моделирование.*

#### **Условие**

Определите сумму количества треугольников, граней, вертексов (`tris + edges + verts`).

#### **Решение**

В большинстве 3D-редакторов можно включить и настроить отображение счетчика этих трех элементов, сложить три целых числа на калькуляторе.

**Ответ:** 561.

### **Задача 3.2.1.5. Одна нормаль (5 баллов)**

*Темы: арт, 3D-моделирование.*

#### **Условие**

Внимательно осмотрите модель и сосчитайте количество вертексов, которые имеют всего одну нормаль для всех прилегающих полигонов/граней (иначе говоря, имеют тип нормали `unify`).

#### **Решение**

В большинстве 3D-редакторов есть инструменты редактирования нормалей, которые позволяют увидеть их. Однако это возможно определить и на глаз по шейдингу, но очень сложно. Два искомых вертекса визуально спрятаны внутри другого объекта меша.

**Ответ:** 5.

### **Задача 3.2.1.6. Спрятанные цифры (10 баллов)**

*Темы: арт, графический редактор.*

#### **Условие**

Извлеките из файла текстуру и, используя графический редактор, отредактируйте изображение так, чтобы проявился спрятанный там код.

### ***Решение***

Путем изменения насыщенности. Первая и последняя цифры кода выходят на соседние цвета.

**Ответ:** 543262.

### ***Программирование***

#### ***Задача 3.2.1.7. Массивы (20 баллов)***

Темы: программирование, обработка массивов.

### ***Условие***

Даны три массива  $A$ ,  $B$  и  $C$  длины  $n$ , нумерация элементов которых ведется с 1, а также числа  $x_0$  и  $y_0$ . Построим два массива  $X$  и  $Y$  длины  $(n + 1)$  по следующим правилам:

- $X_0 = x_0$ ,  $Y_0 = y_0$ .
- Для всех  $i$  от 1 до  $n$  выполним следующие действия:
  1. зададим элементы, как  $X_i = X_{i-1} + A_i$  и  $Y_i = Y_{i-1} + A_i$ ;
  2. выберем ровно одну из операций ниже и применим ее:
    - ◊  $X_i = \min(X_i, B_i)$ ;
    - ◊  $Y_i = \min(Y_i, C_i)$ .

Построим массивы  $X$  и  $Y$  так, чтобы максимизировать значение  $X_n + Y_n$ .

Найдите максимальное значение  $X_n + Y_n$ , которое можно получить, выполняя описанные выше действия.

### ***Формат входных данных***

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 100\,000$ ) — длина массивов  $A$ ,  $B$  и  $C$ .

Во второй строке даны  $n$  целых чисел  $A_1, A_2, \dots, A_n$  ( $0 \leq n \leq 10^9$ ) — массив  $A$ .

Во второй строке даны  $n$  целых чисел  $B_1, B_2, \dots, B_n$  ( $0 \leq n \leq 10^9$ ) — массив  $B$ .

Во второй строке даны  $n$  целых чисел  $C_1, C_2, \dots, C_n$  ( $0 \leq n \leq 10^9$ ) — массив  $C$ .

В пятой строке даны два целых числа  $x_0$  и  $y_0$  ( $0 \leq x_0, y_0 \leq 10^9$ ).

### ***Формат выходных данных***

Выведите единственное число — максимально возможное значение  $X_n + Y_n$  среди всех возможных вариантов построить массивы  $X$  и  $Y$ .

## **Решение**

Тесты и решение: <https://disk.yandex.ru/d/UJpU7ap5H27EcQ>.

### **Задача 3.2.1.8. Графы (20 баллов)**

Темы: программирование, алгоритмы, графы.

#### **Условие**

В данной задаче дано два графа. Всего в первом графе есть  $n$  вершин, притом в графе есть  $m_1$  дорог,  $i$ -я из них соединяет вершины  $u_i$  и  $v_i$ . Во втором графе также есть  $n$  вершин, притом в графе есть  $m_2$  дорог,  $i$ -я из них соединяет вершины  $a_i$  и  $b_i$ . Кроме того, второй граф является связным.

Доступна функция преобразования, позволяющая выбрать любые две различных вершины  $x$  и  $y$  и изменить смежность между ними в обоих графах. То есть если в одном графе между вершинами  $x$  и  $y$  не было ребра, функция добавит ребро, а если ребро было, функция его уберет. При этом функция изменяет ребра одновременно для обоих графов (для каждого графа по отдельности).

Задача: использовать эту функцию не больше, чем  $n$  раз так, чтобы первый граф стал связным. При этом после каждого применения функции второй граф должен оставаться связным.

Определите, можно ли этого достигнуть, и если да, то найдите подходящую последовательность использования функции.

#### **Формат входных данных**

Каждый тест состоит из нескольких наборов входных данных. В первой строке находятся два целых числа  $t$  и  $g$  — количество наборов входных данных и номер группы тестов,  $1 \leq t \leq 60\,000, 0 \leq g \leq 10$ .

Далее следуют описания наборов входных данных. В первой строке описания каждого набора входных данных находятся три целых числа  $n$ ,  $m_1$  и  $m_2$  — количество вершин, количество ребер в первом графе и количество ребер во втором графе:  $3 \leq n \leq 300\,000; 0 \leq m_1, m_2 \leq 300\,000; m_1, m_2 \leq \frac{n(n-1)}{2}$ .

В следующих  $m_1$  строках содержатся описания ребер первого графа. В  $i$ -й строке находятся два целых числа  $u_i$  и  $v_i$  — номера вершин, соединенных  $i$ -й освещенной дорогой,  $0 \leq u_i, v_i \leq n$ . Гарантируется, что все дороги различны.

В следующих  $m_2$  строках содержатся описания ребер второго графа. В  $i$ -й строке находятся два целых числа  $a_i$  и  $b_i$  — номера городов, соединенных  $i$ -й освещенной дорогой,  $0 \leq a_i, b_i \leq n$ . Гарантируется, что все ребра различны, и что во втором графе между любыми двумя вершинами существует путь.

## **Формат выходных данных**

Для каждого набора входных данных выведите **No**, если не существует последовательности применения функции, удовлетворяющей всем условиям. В противном случае выведите **Yes**.

Во второй строке выведите целое число  $k$  — количество использованных применений функции,  $0 \leq k \leq n$ .

Далее выведите  $k$  строк. В  $i$ -й из них выведите два целых числа  $x_i$  и  $y_i$  — номера городов, к которым применяется  $i$ -я функция,  $1 \leq x_i, y_i \leq n, x_i \neq y_i$ . Обратите внимание, что после каждого применения функции второй граф должен оставаться связным.

## **Решение**

Тесты и решение: <https://disk.yandex.ru/d/5MYfi8HBDLZ1oA>.

## **Продукт**

### **Задача 3.2.1.9. Оценка жизненного цикла игры (5 баллов)**

*Темы: продукт, жизненный цикл игры.*

## **Условие**

Компания разработала мобильную игру, которая будет получать обновления и улучшения в течение года.

Рассчитайте чистую прибыль от игры за 12 месяцев с учетом нескольких факторов. Ответ дайте в рублях.

Сотрудники компании сделали таблицу 3.2.7.

Таблица 3.2.7

Месяц	Число работников в компании	Объем продаж	Маркетинговые расходы	Количество активных пользователей
<b>Январь</b>	70	15 000	1 200 000	12 000
<b>Февраль</b>	65	18 000	1 000 000	15 000
<b>Март</b>	68	20 000	900 000	18 000
<b>Апрель</b>	69	22 000	850 000	20 000
<b>Май</b>	69	25 000	800 000	22 000
<b>Июнь</b>	69	30 000	1 000 000	24 000
<b>Июль</b>	70	28 000	1 100 000	25 000
<b>Август</b>	75	26 000	1 200 000	22 500
<b>Сентябрь</b>	74	24 000	1 300 000	21 000
<b>Октябрь</b>	74	22 000	1 400 000	20 000

Месяц	Число работников в компании	Объем продаж	Маркетинговые расходы	Количество активных пользователей
Ноябрь	80	20 000	1 500 000	19 000
Декабрь	80	18 000	1 600 000	18 000

## Данные

- Цена игры в магазине — 400 руб.
- Объем продаж игры представлен в таблице.
- Доход от встроенных покупок: средний доход от одного игрока от встроенных покупок — 600 руб. 10% игроков делают покупки в игре.
- Количество активных пользователей в месяц (по данным аналитики) находится в таблице.
- Маркетинговые расходы находится в таблице.
- Расходы на поддержку игры (сервера, обновления, техподдержка): 500 000 руб. в месяц.
- Дополнительные единовременные расходы — 3 000 000 руб. за весь год.

## Решение

Общий доход от продаж игры:

$$268\,000 \cdot 400 = 107\,200\,000 \text{ руб.}$$

Доход от встроенных покупок (равен количеству активных пользователей, умноженному на процент игроков, совершивших покупки, умноженному на средний доход от встроенных покупок):

$$236\,500 \cdot 10 \cdot 600 = 14\,190\,000 \text{ руб.}$$

Общий доход (равен доходу от продаж, суммированному с доходом от встроенных покупок):

$$107\,200\,000 + 14\,190\,000 = 121\,390\,000 \text{ руб.}$$

Общие расходы (равны сумме на маркетинг, поддержку и разовые расходы):

$$13\,850\,000 + 6\,000\,000 + 3\,000\,000 = 22\,850\,000 \text{ руб.}$$

Чистая прибыль:

$$121\,390\,000 - 22\,850\,000 = 98\,540\,000 \text{ руб.}$$

**Ответ:** 98 540 000.

## 3.2.2. Командные задачи

Командные задачи второго этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/69924/enter/>.

Команды могут приступить к решению командной части второго тура, а именно — к созданию игрового прототипа. В рамках задания необходимо разработать прототип игры, который в дальнейшем подлежит доработке на заключительном этапе. Разработка должна соответствовать заданному жанру, сеттингу и основной идеи.

**Жанр:** симулятор.

**Сеттинг:** «Берлога».

## *Основная идея игры*

- Управление ресурсами, развитие общин и исследование других миров.
- Симуляция отдельных процессов или действий, например, добыча меда или жизнь отдельно взятого медведя.

## *Примеры игр, которые могут использоваться в качестве референсов*

- SimCity.
- Microcivilization.
- Thronefall.
- Timberborn.

## *Сеттинг «Берлога»*

«Берлога» — игровая платформа, на базе которой выпускается серия свободно распространяемых мобильных игр, объединенных общим сюжетом. Проект представляет собой точку входа для школьников в технологическое образование, способствует их вовлечению в сложное технологическое содержание, привлекает в кружки и инженерные соревнования, а также обучает программированию и другим востребованным навыкам.

## *Основные положения сеттинга*

- **Главные персонажи** — антропоморфные медведи различных видов, проживающие в сложном технологическом мире.
- **Традиции** — общество медведей разделено на касты, каждая из которых выполняет определенную роль (пасечники, конструкторы, биоинженеры и др.).
- **Ресурсы** — мед является основным топливом и ресурсом, необходимым для создания биотоплива, используемого в путешествиях и технологиях.
- **Кротовые норы** — перемещение по Вселенной осуществляется через особые участки пространства с использованием сложных расчетов и биотоплива.
- **Общинный образ жизни** — медведи живут группами, редко действуя в одиночку, и следуют традиционным правилам.
- **Экономика** — понятия «заем» и «долг» отсутствуют, существует только понятие «дар».

- **История** — точные сведения о прошлом медведей отсутствуют.
- **Технологический уровень** — технологии развиты, но основаны на земных аналогах; сверхсветовые гипердвигатели отсутствуют.

Подробное описание сеттинга: <https://disk.yandex.ru/i/2lseC7OsE3KhLA>.

Вики «Берлоги»: [https://berloga.fandom.com/ru/wiki/Berloga\\_Вики](https://berloga.fandom.com/ru/wiki/Berloga_Вики).

## *Основные критерии оценки*

- В игре присутствует персонаж в виде антропоморфного медведя.
- Жанр игры соответствует жанру симулятора.
- В игровом процессе отсутствуют элементы, противоречащие сеттингу «Берлоги».
- В игре отражены ключевые особенности сеттинга.
- Возрастное ограничение игры — 12+.

## *Модульная структура разработки*

Для успешной разработки прототипа игровые задания разделены на несколько тематических блоков (модулей). Каждый из них посвящен отдельному аспекту проекта и содержит декомпозиционные задачи, выполнение которых позволяет поэтапно разрабатывать собственный прототип.

В модуле «Продукт» оценивается итоговый билд игры и успешность общей композиции всех реализованных решений. Для получения максимального балла необходимо выполнить задания каждого модуля.

### ***Задача 3.2.2.1. Геймдизайн (100 баллов)***

*Темы: знание инструментов для разработки игр, навык сценарного проектирования, знание и умение проектирования игровых механик, интерфейс, умение анализировать игры, геймдизайн, игровые механики, геймкрафт.*

Геймдизайн является одним из наиболее комплексных блоков заданий. Принятые в нем решения оказывают первостепенное влияние на итоговую игру. В рамках второго этапа необходимо реализовать прототип игры.

Перед началом работы рекомендуется учесть следующие аспекты:

- Вначале следует **ознакомиться со всеми пунктами**. Несмотря на их большое количество, предварительное изучение позволит определить наиболее удобный порядок оформления решений.
- **Порядок пунктов изменять не рекомендуется**. Это позволит экспертам быстрее и качественнее провести проверку, что снизит вероятность необходимости подачи апелляции, результаты которой могут повлиять на итоговый балл.
- **Использование схем и таблиц** способствует сокращению объема текста и облегчает процесс проверки экспертами.

- **Избыточное описание нежелательно.** Концепт-документ должен содержать краткое и информативное изложение сути проекта, позволяя быстро понять основные аспекты игры и ее разработки. В случае сомнений относительно необходимости сокращения текста рекомендуется провести тестирование на членах команды. Эксперты испытывают значительную нагрузку при проверке работ, поэтому лаконичность и ясность изложения являются приоритетными.

## **Структура концепт-документа. Техническое задание**

### ***Титульный лист***

Укажите название команды, название игры (красивый арт, связанный с игрой, одобряется).

### ***Краткое описание игры***

Напишите краткое описание игры по методологии gamecraft (опционально):  
<https://disk.yandex.ru/i/reXoqAweY5pZPw>.

Описание должно включать в себя: жанр, сеттинг, предмет симуляции, ЦА, платформу.

Результат: выжимка концепта игры, 200 символов максимум.

### ***Анализ жанра***

Симулятор — довольно обширный жанр. Предлагаем его уточнить. Симулятивность может проявляться в различных аспектах игры. Опишите, на какой механике акцентируете внимание. Быть может это экономический симулятор, где идет имитация процессов фондового рынка. Или управление и ремонт дронов. Конкретизируйте, что симулирует игра или с каким жанром сочетается. Поясните логичность своей идеи.

### ***Фичлист (feature list)***

Фича (англ. feature) — единица контента, при добавлении которой в игру она (игра) становится заметно отличной от предыдущей версии.

Примеры:

- при добавлении системы инвентаря игра становится значительно иной, появляются дополнительные пласти геймплея (менеджмент инвентаря, и т. д.);
- добавление системы диалогов — фича, добавление одного диалога при наличии других — нет.

Создайте таблицу, где будут описаны все фичи в проекте и разъяснение того, что означает готовность фичи (в том числе количественный эквивалент).

Старайтесь зафиксировать этот список фич и во время разработки двигаться по нему.

Не добавляйте слишком много тяжелых фич, здраво оценивайте свои силы (а если добавили, докажите свои навыки уже на этом этапе хорошим прототипом).

Пример выполнения описан в таблице 3.2.8.

Таблица 3.2.8

Фича	Определение готовности
<b>Управление</b>	WASD+Space, вращение камерой
<b>Перемещение</b>	4 направления, прыжок, высота прыжка [edit], скорость передвижения [edit]
<b>Сварка</b>	Нанесение шва на поверхность, скрепление предметов, максимальный вес, который может выдержать шов [edit]
<b>Локации</b>	3 локации (дом, мастерская, рынок), дом имеет процедурную генерацию
<b>Система повреждений</b>	Предметы могут разрушаться по фиксированным паттернам, мелкие элементы уничтожаются без возможности восстановления
и т. д.	и т. д.

## Анализ референсов

Выберите три разные игры, которыми вдохновляетесь при создании проекта.

Среди референсов должны быть:

- **Визуальный референс** — игра со схожим визуальным стилем, поясните выбор и то, чем схожа ваша игра именно с этой игрой.
- **Геймплейный референс** — откуда почерпнули некоторые механики, на что опирались при продумывании core/meta циклов. Опишите, что взяли из проекта и почему.
- **Рыночный референс** — выберите игру со схожей целевой аудиторией, платформами, целью. Опишите схожесть (**совет**: игры «Берлоги» потенциально могут быть хорошими рыночными референсами).

От описания референса требуется:

- Изображение игры, подходящее под пункт референса (для рыночного можно обложку).
- Название и ссылка (в случае мультиплатформы предпочтителен Steam).
- Почему целевая аудитория схожа с вашей (кратко).
- Опишите, что конкретно взяли из данной игры: например, конкретную механику, сюжет, либо визуал.
- Почему именно эта игра (ответы похожие на: «любимая игра детства», «художник иначе не умеет» — не подходят, требуется более конструктивный ответ).

## Анализ конкурентов

Выберите две игры в том же поджанре и с пересекающейся платформой (RDR2 — не конкурент Candy Crash) и аудиторией.

Источники представьте в виде ссылки на страницу, где можно подтвердить количественную характеристику (в случае большого объема информации на странице и если написания указанного числа недостаточно, чтобы найти нужную информацию, пожалуйста, укажите ключевые слова для поиска).

От описания конкурента требуется:

- изображение (обложка/скриншот вполне подходят);
- название и ссылка (в случае мультиплатформы предпочтителен Steam);
- бюджет, в случае именитых игр (с источником);
- количество установок (с источником), считаем по количеству скачиваний;
- причины, почему целевая аудитория схожа с вашей (кратко).

## ***Паспорт игры***

Паспорт игры — это таблица, где описана минимально необходимая информация об игре, чтобы сложить о ней хоть какое-то мнение. Ниже представлена таблица, которую нужно заполнить.

Таблица 3.2.9

<b>Название</b>	<b>Название игры</b>
Сегмент + длина сессии	Casual/Midcore/Hardcore Длина сессии
Жанр	Жанр/комбинация
Платформы	Опишите платформы, на которые в теории будет релизироваться игра (ПК/смартфоны (указать магазины), консоли (указать модель))
Целевая аудитория	Опишите ЦА (тезисно, 2–3 пункта),
Сеттинг	Конкретизируйте игру в мире «Берлоги» (кратко); можете указать планету/событие
Монетизация	Напишите название модели монетизации (учтывайте, что находитесь в контексте НКФП «Берлога»)
Технологическая платформа	Напишите движок и его версию
Стилистика	Опишите тезисно, как игрок видит игру; например: «3D от третьего лица. Middle-poly»
Основная идея сюжета	В одном предложении опишите идею сюжета; например: «Задача Руки от нападений нечистой силы во главе с темным Ермаком»

## ***Целевая аудитория***

Важно понимать, для кого разрабатывается проект. В интернете популярно мнение, что настоящего геймдизайнера от совсем новичка отличает то, что новичок делает игру для себя, а геймдизайнер — для аудитории. Поэтому, чтобы лучше понимать потребности и принимать верные решения, необходимо хорошо проанализировать целевую аудиторию.

## ***Описание целевой аудитории (ЦА)***

Приведите описание целевой аудитории, используя географические, социально-демографические и поведенческие критерии. Приведите смежные жанры, которыми еще может быть заинтересована ЦА.

### ***Как понимать «поведенческие критерии»?***

Нужно указать описание увлечений и какие-то аспекты характера, на которые можно ориентироваться (то есть не милый-позитивный, а азартный, экстраверт и т. п.). Также вместо аспектов характера хорошо подойдут психотипы по Бартлу, пример:

- увлекается настольными играми европейской школы;
- увлекается математикой и психологией;
- планировщик по расширенным психотипам Бартла.

## ***Анализ ЦА***

Дайте ответы на эти вопросы, чтобы лучше понимать, какие решения предпринимать и в какую сторону масштабировать проект перед финалом:

- Какие потребности и ожидания есть у целевой аудитории?
- В каких чувствах и эмоциях нуждается аудитория?
- Какой уровень сложности и вызовы предлагаете игрокам?
- Какую идею или посыл хотите донести до игроков?

## ***Сюжет***

Опишите сюжет в игре. В начале ответьте, какую функцию он выполняет в игре, затем опишите основные события, персонажей (лучше всего представить в виде схемы с дополнительным пояснением пунктов).

## ***Краткая выжимка из сюжета***

Напишите сводку сюжета (опишите события максимально тезисно с небольшими описаниями). Один из оптимальных путей — использование нейросетей (**и редактура после них**). Описание **до 500** символов.

## ***Главный герой***

Опишите главного героя, чем он выделяется на фоне остальных, какие мотивы им движут, есть ли сверхзадача (если да, то какая), опишите характер, добавьте изображение (или лаконичное описание) внешности. Опишите компаньонов героя и их функционал, в случае их отсутствия поясните причину.

## ***Атмосфера игры***

Опишите желаемую атмосферу (какие эмоции и чувства должна вызывать игра) и подумайте, какие механики и иные решения могут обеспечить ее.

## ***Инструменты повествования***

Опишите, каким образом будет производиться повествование сюжета и лора:

- Какие приемы будут для этого использоваться? Если это будут записки или схожие явления, то как они будут реализованы?
- В каком объеме и с какой целью будут подаваться игроку? Если это будут видео или кат-сцены, то как и с какой целью они будут показываться?
- В каком формате они будут реализованы: видеоролики, флэш-анимация, внутриигровые сцены на движке игры, слайды с картинками? Для более точного пояснения можно применять ссылки к другим играм, фильмам, мультфильмам и комиксам.

## ***Описание геймплея***

Охарактеризуйте общий принцип геймплея, а также наиболее интересные особенности игры. Не вдавайтесь в подробности и представьте их как можно более привлекательно. Объем текста — не более 1 200 символов.

## ***Сводка геймплея***

Опишите главный концепт действий максимально кратко, очень желательно продемонстрировать с помощью игровых циклов и User Story.

В случае описания текстом рекомендуем использовать нейросети для сокращения текста, который должен содержать до 500 символов.

## ***Core и meta***

Охарактеризуйте по две механики из двух core- и двух meta- циклов, при этом ответьте на вопросы:

- Каков принцип действия механики (кратко расскажите о том, как механика влияет на геймплей)?
- Какой опыт она обеспечивает для игрока?
- Как она потенциально комбинируется с другими механиками цикла (хотя бы две комбинации; не обязательно использовать механики, описанные в этом задании)?

## ***Проработка геймплея***

Необходимо прописать особенные геймплейные механики игры, объяснить логику их работы.

Следует обосновать их приемлемость в рамках игровой концепции: насколько и почему они вписываются в сеттинг, жанр, какие решает задачи с точки зрения геймплея.

Также нужно пояснить, какие возможности механика предоставляет игроку, какие его потребности она закрывает и насколько она ему актуальна.

Требуется прописать три механики.

Таблица 3.2.10

Механика	Приемлемость механики в рамках игровой концепции — какие задачи и проблемы решает механика	Каким образом игра закрывает потребности игрока?	Ссылка на видео с показом механики, залитое на видеохостинг/облачо (удостоверьтесь, что доступ есть у всех пользователей)

### ***Игровая сложность. Вызовы и препятствия.***

Какие вызовы ждут игрока? В чем состоит сложность? Насколько должно быть сложно игроку? Опишите опасности, причины возможных ошибок игрока, штрафы за ошибку.

### ***Прогрессия и реиграбельность***

С помощью чего идет развитие персонажа (и самого игрока)? Сколько времени необходимо на достижение всего контента? Какой повод оставаться в игре дальше?

### ***Критерии оценивания***

Оценивается заполненность и смысловая связь ответа с описанием раздела.

Критерии	Балл
Краткое описание игры (оценивается только в случае соответствия количеству символов)	5
Жанр	1
Сеттинг	1
Предмет симуляции	1
ЦА	1
Платформа	1
<b>Анализ жанра</b>	<b>2</b>
Комбинация/аспект симуляции выбран	1
Выбор качественно обоснован	1
<b>Фичлист</b>	<b>15</b>

<b>Критерии</b>	<b>Балл</b>
<i>Описано от 10 до 35 фич</i>	<b>15</b>
Описано от 10 до 35 фич	5
Описано от 10 до 35 фич и дано логичное и исчерпывающее описание каждой	5
Описано от 10 до 35 фич и в проекте нет фич, которые не описаны в фичлисте	5
<i>Описано более 35 фич</i>	<b>15</b>
Описано более 35 фич, дано логичное и исчерпывающее описание каждой, в билде качественно реализована значительная часть этих фич	15
<b>Анализ референсов</b>	<b>6</b>
<i>Визуальный референс</i>	<b>2</b>
Описана схожесть ЦА и то, что взято	1
Заполнены остальные пункты	1
<i>Геймплейный референс</i>	<b>2</b>
Описана схожесть ЦА и то, что взято	1
Заполнены остальные пункты	1
<i>Рыночный референс</i>	<b>2</b>
Описана схожесть ЦА и то, что взято	1
Заполнены остальные пункты	1
<b>Анализ конкурентов</b>	<b>4</b>
<i>Конкурент №1</i>	<b>2</b>
<b>Взята маленькая по масштабу игра</b>	<b>2</b>
Взята маленькая по масштабу игра; описан бюджет (в случае популярной игры) и количество скачиваний (информация легко считывается из источника)	1
Остальные пункты заполнены	1
<b>Взята масштабная игра</b>	<b>2</b>
Взята масштабная игра; описан бюджет и количество скачиваний (информация легко считывается из источника) и заполнены остальные пункты. Игра на этапе прототипа является конкурентоспособной к приведенной игре)	2
<i>Конкурент №2</i>	<b>2</b>
<b>Взята маленькая по масштабу игра</b>	<b>2</b>
Взята маленькая по масштабу игра; описан бюджет (в случае популярной игры) и количество скачиваний (информация легко считывается из источника)	1
Остальные пункты заполнены	1
<b>Взята масштабная игра</b>	<b>2</b>
Взята масштабная игра; описан бюджет и количество скачиваний (информация легко считывается из источника) и заполнены остальные пункты. Игра на этапе прототипа является конкурентоспособной к приведенной игре)	2
<b>Паспорт игры</b>	<b>5</b>
Заполнено 2 и более пунктов	1
Заполнено 4 и более пунктов	1
Заполнено 6 и более пунктов	1
Заполнено 8 и более пунктов	1
Заполнены все пункты	1
<b>Описание ЦА</b>	<b>2</b>

<b>Критерии</b>	<b>Балл</b>
Описаны географические, социально-демографические и поведенческие критерии	1
Приведены смежные жанры, которыми заинтересована ЦА	1
<b>Анализ ЦА</b>	<b>8</b>
Какие потребности и ожидания есть у целевой аудитории?	2
В каких чувствах и эмоциях нуждается аудитория?	2
Какой уровень сложности и вызовы предлагаете игрокам?	2
Какую идею или посыл хотите донести до игроков?	2
<b>Сюжет</b>	<b>4</b>
Представлена в виде схемы с объяснениями	2
Основные персонажи	1
Основные события	1
<b>Краткая выжимка из сюжета (оценивается только в случае соответствия количеству символов)</b>	<b>3</b>
События написаны тезисно и с небольшим описанием. Соответствует сюжету, описанному выше	3
<b>Главный герой и Ко</b>	<b>2</b>
Описан главный герой	1
Описаны компаньоны/обосновано их отсутствие	1
<b>Атмосфера игры</b>	<b>2</b>
Описаны эмоции игрока	1
Описаны механики достижения атмосферы	1
<b>Инструменты повествования</b>	<b>1</b>
Описаны инструменты повествования и их цель	1
<b>Описание геймплея (оценивается только в случае соответствия количеству символов)</b>	<b>2</b>
Виу-эффект (описания механик не уходит в технический язык, после прочтения появляется желание поиграть)	1
Описание исчерпывающее, не возникает значительных вопросов по концепту геймплея. Из прочтения можно понять, как происходит игровой процесс в общих чертах	1
<b>Сводка геймплея (оценивается только в случае соответствия количеству символов)</b>	<b>3</b>
Дано описание геймплея	1
Из описания можно примерно представить действительный игровой процесс	1
Краткое описание не противоречит большому	1
<b>Core и Meta механики</b>	<b>12</b>
<b>Core №1</b>	<b>3</b>
Описан принцип действия механики	1
Описан опыт игрока от взаимодействия с механикой	1
Описаны как минимум 2 возможные комбинации с другими механиками	1
<b>Core №2</b>	<b>3</b>
Описан принцип действия механики	1
Описан опыт игрока от взаимодействия с механикой	1
Описаны как минимум две возможные комбинации с другими механиками	1
<b>Meta №1</b>	<b>3</b>

<b>Критерии</b>	<b>Балл</b>
Описан принцип действия механики	1
Описан опыт игрока от взаимодействия с механикой	1
Описаны как минимум 2 возможные комбинации с другими механиками	1
<i>Meta №2</i>	<b>3</b>
Описан принцип действия механики	1
Описан опыт игрока от взаимодействия с механикой	1
Описаны как минимум 2 возможные комбинации с другими механиками	1
<b>Проработка Геймплея</b>	<b>21</b>
<i>Механика №1</i>	<b>7</b>
Механика вписывается в сеттинг, жанр и общий геймплей	1
Механика решает задачи, поставленные контекстом игры	2
Механика закрывает потребности игрока, она актуальна	2
Соответствие видео механики ее описанию	2
<i>Механика №2</i>	<b>7</b>
Механика вписывается в сеттинг, жанр и общий геймплей	1
Механика решает задачи, поставленные контекстом игры	2
Механика закрывает потребности игрока, она актуальна	2
Соответствие видео механики ее описанию	2
<i>Механика №3</i>	<b>7</b>
Механика вписывается в сеттинг, жанр и общий геймплей	1
Механика решает задачи, поставленные контекстом игры	2
Механика закрывает потребности игрока, она актуальна	2
Соответствие видео механики ее описанию	2
<b>Игровая сложность</b>	<b>2</b>
Описана концепция сложности, в чем именно состоит челлендж	1
Описаны причины ошибок игрока, цена ошибки, штрафы за совершение ошибки	1
<b>Прогрессия и реиграбельность</b>	<b>1</b>
Описано, с помощью чего идет развитие персонажа. Даны примерные числа на длину геймплея. Описаны механики для удержания игрока	1
<b>ИТОГ:</b>	<b>100</b>

### **Задача 3.2.2.2. Арт (100 баллов)**

Темы: сценарное мастерство, раскадровка, дизайн персонажей, колористика, VFX, Sound дизайн, UI/UX, нейросети, работа со светом, анимация.

Задание поможет декомпозировать визуал игры на ART-задачи, которые в дальнейшем команде будет проще воплощать в прототипе игры.

#### **Сюжет – Конфликт – Нarrатив**

Очень кратко опишите основную идею игры и сюжетный конфликт.

---

Форма: документ с именем: **названиекоманды\_task1art.docx** (от 200 до 600 символов).

### ***Набросок «скриншота»***

Нарисуйте схематический драфт того, как будет выглядеть игра, что будет видеть игрок.

Постарайтесь определить положение камеры, взаимное расположение и размер персонажей относительно камеры.

Форма: строго четыре изображения с названиями:

- **названиекоманды\_task2art\_1.png**,
- **названиекоманды\_task2art\_2.png**,
- **названиекоманды\_task2art\_3.png**,
- **названиекоманды\_task2art\_4.png**.

Разрешение изображения  $1280 \times 720$  px (пикселей).

### ***Дизайн главного героя***

Разработайте образ и внешний вид главного героя, изобразите его в разных позах и состояниях.

Форма: документ с именем: **названиекоманды\_task3art.png**.

Разрешение изображения  $1920 \times 1080$  px (пикселей).

### ***Колористика***

Изобразите цветовую палитру игры. Рекомендуется использовать цветовой круг (колесо).

Форма: документ с именем: **названиекоманды\_task4art.png**.

Разрешение изображения  $512 \times 512$  px (пикселей).

### ***VFX***

Продемонстрируйте примеры визуальных эффектов, которые будут использованы в игре.

Форма: документ с именем: **названиекоманды\_task5art.mp4**.

Размер файла до 5 MB.

### ***Подбор текстур***

Создайте коллаж из квадратных текстур, которые могут быть использованы для оформления локаций в игре.

Форма: документ с именем: **названиекоманды\_task6art.png**.

---

Разрешение изображения 2 048 × 2 048 px (пикселей).

### ***Аудио оформление***

Предоставьте демо шумов и отрывки музыкальных тем.

Форма: документ с именем: **названиекоманды\_task7art.mp4**.

Размер файла до 5 MB.

### ***Му́дборд***

Соберите коллаж из изображений, передающих настроение игры и формирующих визуальный ее стиль.

Форма: документ с именем: **названиекоманды\_task8art.png**.

Разрешение изображения 1 920 × 1 080 px (пикселей).

### ***UI/UX***

Соберите коллаж из UI-элементов, которые будут использованы в игре.

Форма: документ с именем: **названиекоманды\_task9art.png**.

Разрешение изображения 1 920 × 1 080 px (пикселей).

### ***Использование нейросетей***

Используйте нейросети, чтобы сгенерировать контент, который мог бы быть использован в игре (референсы, текстуры).

Форма: документы с именами:

- **названиекоманды\_task10art\_1.png**,
- **названиекоманды\_task10art\_2.png**,
- **названиекоманды\_task10art\_3.png**,
- **названиекоманды\_task10art\_4.png**.

Разрешение изображения 512 × 512 px (пикселей).

### ***Свет***

Изобразите, как свет будет использоваться в игре (коллаж из референсов световых решений, схема освещения).

Форма: документ с именем: **названиекоманды\_task11art.png**.

Разрешение изображения 1 920 × 1 080 px (пикселей).

## **Анимация**

Создайте gif-изображение с зацикленной анимацией, которое могло бы быть использовано в игре или сообществе для продвижения (покадровая, 3D-рендер, VFX).

Форма: документ с именем: **названиекоманды\_task12art.gif**.

Размер файла до 2 МВ. Разрешение 256 × 256 px (пикселей).

## **Критерии оценивания**

<b>Критерии</b>	<b>Балл</b>
<b>Сюжет — Конфликт — Нarrатив</b>	<b>5</b>
Наличие сюжетного конфликта	2
Позитивная идея (сверхзадача)	2
Есть развитие сюжета	1
<b>Набросок «скриншота»</b>	<b>12</b>
Есть условные UI-изображения	2
Есть цвет/свет/тон	3
Есть изображения игровых / неигровых персонажей	4
Есть изображения окружения	3
<b>Дизайн главного героя</b>	<b>15</b>
Персонаж из сеттинга	2
Костюм	2
Освещение, тени	2
Детализация персонажа (дополнительные элементы)	2
Эмоции (мимика)	1
2 ракурса ГГ	1
3 ракурса ГГ	2
Больше 4 ракурсов ГГ	3
<b>Колористика</b>	<b>3</b>
Используется 2–3 тона	1
Использовано сочетание цветов по одной из основных схем	2
<b>VFX</b>	<b>8</b>
Эффекты сочетаются между собой	3
Более 4	3
Более 8	2
<b>Подбор текстур</b>	<b>6</b>
Текстуры гармонируют между собой	3
Более 4 текстур	1
Более 9 текстур	2
<b>Аудиооформление</b>	<b>6</b>
Фоновая музыка	2
Звуки персонажа	1

<b>Критерии</b>	<b>Балл</b>
Звуковые эффекты	2
Звуки окружения	1
<b>Мудборд</b>	<b>3</b>
Есть общее настроение	1
Есть общий визуальный стиль	1
Более 8 изображений	1
<b>UI/UX</b>	<b>14</b>
Более 4 разных элементов	3
Более 9 разных элементов	3
Наличие общего стиля	3
Соответствие настроению игры и цветовому решению	3
Кнопки, курсоры	2
<b>Нейросети</b>	<b>8</b>
Изображения имеют смысл	4
Соответствуют визуальному стилю игры	4
<b>Интерактивные объекты</b>	<b>10</b>
Использование света в окружении	4
Использование световых приемов	4
Использование света в игровых механиках	2
<b>Анимация</b>	<b>10</b>
Зациклено	3
Соответствует стилю проекта	3
Анимация длинная или сложная	4
<b>ИТОГ:</b>	<b>100</b>

### **Задача 3.2.2.3. Программирование (100 баллов)**

Темы: знание инструментов для разработки игр, навык реализации игровых механик, программирование, игровые механики.

#### **Условие**

Интернет окружает нас повсюду, мы пользуемся им постоянно для хранения, передачи, поиска и обработки информации. Согласитесь, было бы грустно, если бы игра не могла взаимодействовать с таким мощным инструментом. В этом задании участники учатся использовать API, предоставленное организаторами, для хранения, просмотра и изменения информации о ресурсах в игре.

Реализация задачи делится на три уровня:

1. Базовая реализация — до 30 баллов.
2. Усложненная реализация — до 60 баллов.
3. Полная реализация — до 100 баллов.

## ***Описание задачи***

Необходимо реализовать механику сохранения и получения ресурсов пользователя и внутриигровых магазинов с помощью запросов к серверу. Участникам предоставлен API с документацией по структуре запросов. Кроме того, необходимо учитывать возможность игры разными игроками, соответственно, сохранение данных должно работать для разных пользователей.

## ***Результат работы***

Команде необходимо продемонстрировать работу механики в игре. Жюри при проверке будет отслеживать изменения и запросы на сервере от игры (будет использовано новое имя пользователя), в случае отсутствия определенных запросов, за них будет поставлено 0 баллов.

## ***Дополнительная информация***

- Инструкция по работе с API: [https://disk.yandex.ru/d/A6pT\\_15jkDgthQ](https://disk.yandex.ru/d/A6pT_15jkDgthQ).
- После окончания второго этапа все записи игроков, магазинов, логов и ресурсов будут стерты, останутся только записи игр, следовательно, игра должна уметь создавать игроков и их магазины сама с нуля, зная только `uuid` игры.
- Распространять и передавать данные доступа к API строго запрещено.
- Апелляции на нулевые баллы **не** будут рассмотрены.
- Подумайте о том, как игра будет работать в случае отсутствия интернета, такие сценарии тоже будут рассматриваться при проверке.

## ***Общие критерии оценки***

- В процессе работы механика присыпает запросы на сервер.
- Запросы на сервер не излишни, а срабатывают только при надобности (если по завершению квеста игрок получает 50 монет, это должно быть реализовано в одном запросе, а не в 50 по одной монете).
- Данные в запросах могут быть интуитивно сопоставлены с событиями в игре (например, запрос «изменить количество в переменной `honeу`» при изменении количества «меда» в игре).

## ***Базовая реализация***

Механика хранит на сервере текущее количество всех изменяемых игровых ресурсов для определенного игрока. В процессе проверки будет наблюдаться изменение значений переменных на сервере при изменении соответствующих значений в игре.

## ***Критерии***

- На сервере появляются значения количества ресурсов, которые использованы в игре.

- Наименование переменных на сервере может быть интуитивно однозначно со-поставлено показаниям количества некоего ресурса в игре.
- На сервере обновляются значения количества ресурсов при каждом их изменении в игре.
- Игра поддерживает возможность сохранения данных для игроков с разными никнеймами.

### ***Усложненная реализация***

Кроме указанного задания, приведенного в первом пункте, необходимо отправлять на сервер логи, в которых будет указано, в результате каких действий на сколько изменилась некая переменная. Также необходимо реализовать систему, при которой важные игровые параметры (количества ресурсов), которые хранятся на сервере, являются заведомо правильными. Для этого следует использовать запросы на получение данных с сервера для получения текущего значения и проводить операции с этими данными, после отправляя их обратно на сервер. Таким образом, вне зависимости от изменения количества ресурсов на локальной машине в текущий момент, при проведении пользователем операций с ресурсом, которые влияют на его количество, показания количества ресурсов на сервере будут изменяться лишь предусмотренным игрой способом.

### ***Критерии***

- На сервере появляются значения количества ресурсов, которые использованы в игре.
- Наименование переменных на сервере может быть интуитивно однозначно со-поставлено показаниям количества некоего ресурса в игре.
- На сервере изменяются значения количества ресурсов, которые изменяются в игре.
- Игра поддерживает возможность сохранения данных для игроков с разными никнеймами.
- На сервере появляются сообщения, которые описывают причину изменения какой-либо из хранимых переменных и само изменение.
- Лог-сообщения позволяют однозначно установить, что некое действие, произведенное жюри в игре, привело к изменению данной переменной (информационность сообщений).
- Изменение локального значения переменных, отвечающих за количество ресурсов, не позволяет изменить количество ресурсов на сервере.

### ***Полная реализация***

Кроме указанного в первом и втором пункте, нужно реализовать магазин внутри игры. В некий момент прохождения игры на сервер отправляются данные о товарах в магазине, а игроку открывается возможность покупать эти предметы за ресурсы. Купленные предметы также необходимо сохранять на сервере аналогично ресурсам. Для получения дополнительных баллов следует реализовать больше одного предмета

в магазине, механику продажи предметов, а также все механики из предыдущих критериев.

### ***Критерии***

- На сервере появляются значения количества ресурсов, которые использованы в игре.
- Наименование переменных на сервере может быть интуитивно однозначно сопоставлено показаниям количества некоего ресурса в игре.
- На сервере изменяются значения количества ресурсов, которые изменяются в игре.
- Игра поддерживает возможность сохранения данных для игроков с разными никнеймами.
- На сервере появляются сообщения, которые описывают причину изменения какой-либо из хранимых переменных и само изменение.
- Лог-сообщения позволяют однозначно установить, что некое действие, произведенное жюри в игре, привело к изменению данной переменной (информационность сообщений).
- Изменение локального значения переменных, отвечающих за количество ресурсов, не позволяет изменить количество ресурсов на сервере.
- Магазин присутствует в игре, в нем есть хотя бы один предмет.
- Данные о предметах в магазине, а также о их стоимости (измеряемой в игровых ресурсах), хранятся на сервере.
- В магазине присутствует, по крайней мере, пять предметов.
- В магазине можно не только покупать, но и продавать предметы.
- На сервер отправляются логи с информацией о покупке/продаже.
- В момент покупки, продажи, а также отображения цен при загрузке магазина информация о количестве необходимых и/или имеющихся в наличии предметов получается и отправляется на сервер путем API-запросов.

### ***Критерии оценивания***

<b>Критерии</b>	<b>Балл</b>
<b>Базовая демонстрация</b>	<b>30</b>
На сервере появляются значения количества ресурсов, которые использованы в игре	6
Наименование переменных на сервере может быть интуитивно однозначно сопоставлено показаниям количества некоего ресурса в игре	4
На сервере обновляются значения количества ресурсов при каждом их изменении в игре	10
Игра поддерживает возможность сохранения данных для игроков с разными никнеймами	10
<b>Усложненная реализация</b>	<b>60</b>
На сервере появляются значения количества ресурсов, которые использованы в игре	6
Наименование переменных на сервере может быть интуитивно однозначно сопоставлено показаниям количества некоего ресурса в игре	4
На сервере изменяются значения количества ресурсов, которые изменяются в игре	10
Игра поддерживает возможность сохранения данных для игроков с разными никнеймами	10

<b>Критерии</b>	<b>Балл</b>
На сервере появляются сообщения, которые описывают причину изменения какой-либо из хранимых переменных и само изменение	10
Лог-сообщения позволяют однозначно установить, что некое действие, произведенное жюри в игре, привело к изменению данной переменной (информативность сообщений)	10
Изменение локального значения переменных, отвечающих за количество ресурсов, не позволяет изменить количество ресурсов на сервере.	10
<b>Полная реализация</b>	<b>100</b>
На сервере появляются значения количества ресурсов, которые использованы в игре	6
Наименование переменных на сервере может быть интуитивно однозначно сопоставлено показаниям количества некоего ресурса в игре	4
На сервере изменяются значения количества ресурсов, которые изменяются в игре	10
Игра поддерживает возможность сохранения данных для игроков с разными никнеймами	10
На сервере появляются сообщения, которые описывают причину изменения какой-либо из хранимых переменных и само изменение	10
Лог-сообщения позволяют однозначно установить, что некое действие, произведенное жюри в игре, привело к изменению данной переменной (информативность сообщений)	10
Изменение локального значения переменных, отвечающих за количество ресурсов, не позволяет изменить количество ресурсов на сервере.	10
Магазин присутствует в игре, в нем есть хотя бы один предмет	10
Данные о предметах в магазине, а также о их стоимости (измеряемой в игровых ресурсах), хранятся на сервере	5
В магазине присутствует, по крайней мере, пять предметов	5
В магазине можно не только покупать, но и продавать предметы	10
На сервер отправляются логи с информацией о покупке/продаже	5
В момент покупки, продажи, а также отображения цен при загрузке магазина, информация о количестве необходимых и/или имеющихся в наличии предметов получается и отправляется на сервер путем API-запросов	5
<b>ИТОГ:</b>	<b>100</b>

### **Задача 3.2.2.4. Продукт (300 баллов)**

Темы: видеодемонстрация, презентация игрового проекта, итоговый билд.

#### **Тизер игры (30 баллов)**

Темы: продукт, тизер игры, презентация игрового проекта.

Создайте короткий и интригующий тизер, который привлечет внимание аудитории и вызовет интерес к игре.

При создании тизера не забудьте, что:

- необходимо добавить название команды и игры в начале тизера, при этом креативность появления приветствуется;
- следует использовать качественные материалы (Full HD ( $1920 \times 1080$ ) и выше);
- нужно подобрать подходящую музыку и звуки, которые передают настроение игры;
- креатив приветствуется;

- продолжительность тизера не должна превышать 30 с.

Необходимо отправить ссылку на видео в качестве решения.

### ***Критерии оценивания***

<b>Критерии</b>	<b>Балл</b>
<b>Качество монтажа</b>	<b>4</b>
Название команды	1
Название игры	1
Соединенные кадры несут главную мысль	2
<b>Качество материала</b>	<b>5</b>
Хорошее качество видео (Full HD ( $1920 \times 1080$ ) и выше)	2
Количество шотов (2–3 — 2 балла, 4 и больше — 3 балла) *не считая заставки	3
<b>Качество демонстрации достоинств игры</b>	<b>3</b>
Сделан акцент на арт	1
Сделан акцент на геймплей	1
Сделан акцент на сюжет	1
<b>Звуковое решение</b>	<b>5</b>
Наличие музыки	2
Звуки из игры, которые придают контраст тизеру	3
<b>Творческое решение (креатив)</b>	<b>10</b>
Творческое решение	5
Увлекательность/динамичность тизера	5
<b>Соответствие целевой аудитории</b>	<b>3</b>
Соответствие целевой аудитории	3
<b>Всего:</b>	<b>30</b>

### ***Презентация игры инвестору (50 баллов)***

*Темы: продукт, презентация игрового продукта.*

Пора презентовать игру заказчику, для этого требуется собрать все наработки в единую презентацию и рассказать про проект. Ниже представлен шаблон для презентации, который необходимо использовать.

Продемонстрируйте инвесторам, что серьезно настроены на успех! Начните вести «Дневник разработчика» на протяжении всего периода второго этапа — инвесторы оценят. Не стесняйтесь делиться новостями об игре: пишите посты на школьном сайте, в популярных каналах и даже в СМИ!

Шаблон презентации: <https://disk.yandex.ru/i/PwUu1W91TNCZ-A>

### ***Критерии оценивания***

<b>Критерии</b>	<b>Балл</b>
<b>Паспорт игры</b>	<b>2</b>
Наличие обложки игры (обложка привлекательна +1)	1
Название, жанр, сеттинг, платформа и аудитория указаны верно (1 балл — частично верно)	1
<b>Игры референсы</b>	<b>3</b>
Насколько правильно проведен анализ референсов и по какому из признаков (механики/визуал)	3
<b>Описание геймплея</b>	<b>2</b>
Дано описание с краткой идеей (1 балл — неважно, 2 балла — нормальное)	2
<b>Описание ЦА</b>	<b>6</b>
Вовлеченность	2
Психотип	2
Возраст	2
<b>Геймдизайн</b>	<b>4</b>
Мотивация игрока (core механики)	2
Описаны механики	2
<b>Арт-концепция + mood board</b>	<b>3</b>
Правильно подобран визуальный стиль	1
Насколько подобранные референсы соотносятся с итоговым стилем игры	2
<b>Почему вы верите в эту идею?</b>	<b>3</b>
Проведен анализ конкурентов, приведены диаграммы, статистика похожих игр и результаты опросов и исследований среди целевой аудитории, экспертные мнения в области, на которую ориентирована идея, анализ трендов и перспектив развития рынка.	2
Прикреплен Telegram-канал игры «Дневник разработчика», либо любые другие статьи об игре в СМИ, на школьном сайте и др.	1
<b>Видео геймплея + билд (свой вкус)</b>	<b>25</b>
Наличие рабочего UI + насколько UI является интуитивным	5
Насколько геймплей погружает игрока (поддержание темпа игры, количество задач на весь геймплей)	5
Насколько анимации способствуют развитию темпа игры (плавность/резкость анимации в зависимости от темпа)	5
Реализация механик как способ заинтересовать игрока	5
Играбельность	5
<b>Команда — портфолио</b>	<b>2</b>
Прикреплено фото/арт участника	1
Приведено по 2 задачи — чем занимался участник команды	1
<b>Всего:</b>	<b>50</b>

### **Полезность. «Берлога» в игре (50 баллов)**

**Темы:** продукт, игровой сеттинг.

Подготовьте презентацию, в которой будет обоснована полезность игры и сеттинг «Берлога». Нужно продемонстрировать, как игра способствует развитию технологических навыков, популяризации школьных предметов и решению актуальных

социальных проблем. Также следует показать, как в нее интегрированы элементы обучения для пользователей.

Шаблон презентации: <https://disk.yandex.ru/d/EHVhMv4lhUOB4w>.

Бренд-платформа НКФП «Берлога»: <https://disk.yandex.ru/i/HqRCX1H0Up47hQ>.

Иконки игр: <https://disk.yandex.ru/d/3P6nsvQCq7d1tA>.

Необходимо отправить в форму файл презентации (формат: pdf, до 50 Мб).

### **Критерии оценивания**

<b>Критерии</b>	<b>Балл</b>
<b>Полезность игры</b>	<b>25</b>
Игра способствует развитию навыка из сферы технологий (программирование, конструирование, моделирование, проектирование и т. д.)	5
Игра способствует популяризации школьных предметов (математика, физика, информатика и т. д.)	5
В игре представлена актуальная для общества проблема (экология, глобальное потепление, урбанизация и т. д.)	5
Интерактивное обучение (игровая среда позволяет игрокам учиться на практике (различные стратегии в игре))	10
<b>Сеттинг «Берлога»</b>	<b>25</b>
Представленность в игре традиций сеттинга «Берлога» (творец, первопроходец, биоинженер и т. д.)	5
Использование ключевых слов сеттинга (медовая экономика, небесная цивилизация, цивилизация Севера, путешествия через кротовые норы, биоинженерия и т. д.)	5
Игра не нарушает принципы мира «Берлоги»	5
Наличие в игре отсылок к персонажам и сюжетам других игр и/или комиксов о «Берлоге» (игры — «Академия», «Защита пасеки», «Академия дронов», «Журнал капитана Тундры»; комиксы — «Киберпасека», «Тайна пропавшего меда», «Опасность с медовой горы», «Отроки двух вселенных», «Винтокрыл для друга» и др.)	5
Игра расширяет мир «Берлоги», дополняет сеттинг	5
<b>Всего:</b>	<b>50</b>

### **Страница на VK Play (70 баллов)**

Темы: продукт, публикация игры на платформе.

Создайте привлекательную страницу для игры на платформе VK Play. Страница должна успеть пройти модерацию. (Модерация может занимать от двух до семи рабочих дней, поэтому не затягивайте с этим заданием!)

Необходимо создать кабинет разработчика: [https://documentation.vkplay.ru/p2p\\_vk/p2p\\_application](https://documentation.vkplay.ru/p2p_vk/p2p_application).

В кабинете могут регистрироваться разработчики с 18 лет. Если участникам еще не исполнилось 18, можно зарегистрировать аккаунт на имя своего наставника, учителя, родителя или любого другого человека старше 18 лет, с которым у команды есть связь.

## ***Критерии оценивания***

<b>Критерии</b>	<b>Балл</b>
<b>Страница VK Play прошла модерацию</b>	<b>70</b>
Страница VK Play прошла модерацию	70
<b>Всего:</b>	<b>70</b>

## ***Демонстрация игрового прототипа (70 баллов)***

*Темы: продукт, игровой прототип.*

Демоверсия должна предоставить целостный игровой опыт, поэтому следует проработать этапы от запуска игры до конца прохождения демонстрационного уровня. В прототипе должен быть геймплей на 5 мин игры.

### **Примерный план демонстрации игры (3 мин)**

- Окно меню и настроек — 5–10 с.
- Обучение в игре — 20–30 с.
- Показ геймплея — 40–60 с.
- Демонстрация игровых механик (кор-геймплей и мета-геймплей) — 30–40 с.
- Дизайн уровней и логика — 30–40 с.
- Завершение геймплея.

Отправьте ссылку на видео в качестве решения.

### **Основные критерии оценивания**

- Визуальное исполнение главного меню игры, способствующее заинтересованности игрока.
- Наличие обучающего контента и подсказок в игре.
- Разнообразие и вариативность геймплея, демонстрация начала, середины и конца игры.
- Наполненность игровых уровней.
- Визуальная готовность проекта: работоспособный и прорисованный UI, готовые 2D-спрайты или 3D-модели с текстурами и шейдерами. Наличие НПС, атмосферности и контрастности игрового мира.
- Реализация кор-геймплея и мета-геймплея.
- Наличие звуков и музыки. В игре должно быть несколько треков с плавным переходом между собой (покажите данный переход в видео). Также не забудьте добавить звуки действий.
- Содержательность игры в плане обучения и установки правил.

## ***Критерии оценивания***

<b>Критерии</b>	<b>Балл</b>
<b>Меню</b>	<b>3</b>
Наличие главного меню	1
Визуал главного меню подогревает интерес к игре	2

<b>Критерии</b>	<b>Балл</b>
<b>Обучение</b>	<b>5</b>
Наличие (через квест, информационные элементы, тренировочную зону)	2
Наличие подсказок	1
Понятность и простота обучения	2
<b>Реализованные игровые механики</b>	<b>8</b>
Разнообразие геймплея	2
Проходимость игры (показаны 3 стадии игры — начало, середина, конец)	2
Поддержание сеттинга	2
Вариативность интеракций (игрок может принять решение)	2
<b>Содержание. Логика игрового уровня</b>	<b>7</b>
Отсутствие пустого пространства	2
Полнота карты	1
Разнообразие неигровых персонажей (НПС)	1
Несколько видов сценариев игры/взаимодействия с объектами	2
Простота навигации по уровню/карте	1
<b>Кор-геймплей и мета-геймплей</b>	<b>9</b>
Целостность геймплея (начало/конец, понятность, управление)	3
Реализация механик по ТЗ	3
Реализация отдельных геймлейных фичей (например: диалог, квест — если есть механика: 1 балл, если грамотно работает: 3 балла)	3
<b>Игровой цикл (показан в видео)</b>	<b>6</b>
Действие (что-то сделали)	2
Обратная связь (за это что-то получили)	2
Прогресс (смогли прокачаться за счет meta-механики)	2
<b>Интерфейс (UI/UX)</b>	<b>6</b>
Визуальный стиль интерфейса соответствует стилю игры	2
Содержательность (привычные игроку иконки: 1, понятность: 1)	2
Отсутствие переизбытка UI-элементов во время игры	2
<b>Стиль игры (визуал)</b>	<b>12</b>
Правильное выполнение анимаций	1
Контрастность игрового мира	1
Свет	2
Сложность графики	2
Атмосфера	2
Эстетика (единообразие)	2
Разнообразие спрайтов/3D-моделей	2
<b>Музыка и звуки</b>	<b>7</b>
Наличие музыки (1–2 трека — 1 балл, $\geqslant$ 3 трека — 2 балла)	2
Качество подбора музыки	1
Плавная смена треков	1
Наличие звуков от предметов (открытие чего-то, взятие предмета)	1

<b>Критерии</b>	<b>Балл</b>
Наличие персонажных звуков (враг, ГГ, НПС)	1
Отсутствие противоречия между звуками и действиями	1
<b>Содержательность игры</b>	<b>4</b>
Учебные действия (игра чему-то может научить)	1
Понятность правил (понятность игроку основных принципов игры)	1
Наличие четких указаний игроку, что делать (ясных и неясных)	1
Отсутствие агрессивности игровой среды (негативное воздействие на психику игрока)	1
<b>Уникальность игры</b>	<b>3</b>
Уникальные механики	1
Уникальная графика (необычный стиль)	1
Уникальная идея	1
<b>Всего:</b>	<b>70</b>

### **Итоговый билд (30 баллов)**

Тема: продукт.

#### **Условие**

Демоверсия игры может представлять собой как набор из уровней, так и один комплексный уровень, главное — это продемонстрировать все игровые механики и полноценный игровой процесс.

Оценка демоверсии зависит:

- от стабильности и производительности (должна работать без вылетов и зависаний);
- от корректности работы звуков и музыки;
- от управления (реализовано интуитивно, механики работают стабильно, игра проходима);
- от элементов интерфейса (работают корректно);
- от наличия меню настроек;
- от наличия меню-паузы;
- от возможности выйти из игры через кнопку **Выход**;
- от увлекательности.

Необходимо загрузить исходный код своего проекта на один из представленных ниже сервисов:

- GitLab;
- GitHub.

Код должен быть в главной ветке (`master` или `main`).

#### **Критерии оценивания**

<b>Критерии</b>	<b>Балл</b>
<b>Стабильность и производительность</b>	<b>6</b>
Игра работает без вылетов и зависаний	2
Наличие сохранения данных	2
Графика и анимация работают плавно и без ошибок	2
<b>Музыка и звуки</b>	<b>4</b>
Музыка работает корректно	2
Звуки работают корректно	2
<b>Управление</b>	<b>6</b>
Управление реализовано интуитивно и понятно	2
Механики работают стабильно и не имеют критических багов	2
Проходимость игры	2
<b>Элементы интерфейса</b>	<b>9</b>
Все необходимые элементы интерфейса на месте и работают корректно (внутри меню, меню-паузы)	2
Эффективность реализации работы элементов интерфейса внутри игры	2
Наличие меню-настроек (исправное изменение громкости звуков и музыки — 1 балл + изменение разрешения экрана — 1 балл (2 балла, если игра адаптирована к не стандартному соотношению сторон, например: 16 : 10, 21 : 9, 4 : 3))	3
Наличие меню-паузы	1
Возможность выйти из игры через кнопку <b>Выход</b>	1
<b>Увлекательность</b>	<b>5</b>
Игра затягивает и вызывает желание играть дальше, наличие интересных элементов, стимулирующих продолжить игру	5
<b>Всего:</b>	<b>30</b>

## 4. Заключительный этап

### 4.1. Работа наставника НТО при подготовке к этапу

На этапе подготовки к заключительному этапу НТО наставник решает две важные задачи: помочь участникам в подготовке к предстоящим соревнованиям и формирование устойчивой и слаженной команды. Заключительный этап требует высокой слаженности, уверенности и глубоких знаний, и наставник становится тем, кто объединяет усилия участников и направляет их в нужное русло.

Наставник помогает участникам:

- разобрать задания прошлых лет, используя официальные сборники, чтобы понять структуру финальных испытаний, типы задач и ожидаемый уровень сложности;
- изучить организационные особенности заключительного этапа, включая формат проведения, регламент, продолжительность и технические нюансы;
- спланировать подготовку — на основе даты начала финала составляется четкий график занятий, в котором распределены темы, практикумы и командные тренировки;
- обратиться (при необходимости) за консультацией к разработчикам заданий по профилю, уточнить, на какие аспекты подготовки следует обратить особое внимание, и получить дополнительные материалы.

Также рекомендуется участие в мероприятиях от организаторов, таких как:

- установочные вебинары и открытые разборы задач;
- хакатоны, практикумы и мастер-классы для финалистов;
- встречи в онлайн-формате, информация о которых публикуется в группе НТО во «ВКонтакте» и в телеграм-чатах профилей.

Наставнику необходимо уделить внимание работе на формированием устойчивой, продуктивной и мотивированной команды:

- **Сплочение команды.** Это особенно актуально, если участники живут в разных городах. Регулярные онлайн-встречи, совместная работа над задачами и неформальное общение помогают наладить доверие и улучшить командную динамику.
- **Анализ ролей.** Наставник вместе с командой определяет, кто за что отвечает, какие задачи входят в зону ответственности каждого участника. Также обсуждаются возможности взаимозаменяемости на случай непредвиденных ситуаций.
- **Оценка компетенций.** Важно определить, какими знаниями и навыками уже обладают участники, а какие необходимо развить. На основе этого формируется индивидуальный и командный план подготовки.

- **Участие в подготовительных мероприятиях от разработчиков профилей.** Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.
- **Практика в формате хакатонов.** Наставник может организовать дистанционные хакатоны или практикумы с использованием заданий прошлых лет и методических рекомендаций из официальных сборников.

Таким образом, наставник становится координатором и моральной опорой команды, помогая пройти заключительный этап НТО с максимальной уверенностью и результатом.

## 4.2. Предметный тур

Задачи третьего этапа предметного тура профиля по информатике открыты для решения. Участие в соревновании доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/72671/enter/>.

### 4.2.1. Информатика. 8–11 классы

#### **Задача 4.2.1.1. Контейнеры (10 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### ***Условие***

Кот Матроскин дорос до высокой должности в Трансагенстве. Бочку он уже не катит, а занимается отправкой грузов контейнерами. В точке отправления скопились  $a$  больших контейнеров и  $b$  малых. Матроскин должен спланировать и заказать необходимое количество платформ для их перевозки (открытых вагонов, предназначенных для перевозки грузов, не боящихся атмосферных воздействий). На одну платформу технически можно поместить либо два больших контейнера, либо один большой и два малых. Иных способов загрузить платформу нет. Так как заказ платформы стоит денег, то экономный Матроскин никогда не отправит недогруженную платформу. Если некоторое количество контейнеров не получится загрузить сейчас, он отправит их на ожидание до следующего заказа.

Требуется помочь Матроскину и подсказать, какое максимальное количество платформ он может заказать, чтобы все платформы были полностью загружены контейнерами для отправки.

#### ***Формат входных данных***

В единственной строке содержатся два целых числа  $a$  и  $b$  через пробел — количество больших и малых контейнеров в точке отправления соответственно,  $0 \leq a, b \leq 100$ .

#### ***Формат выходных данных***

Вывести в ответ одно число — максимальное возможно количество платформ, которые получится полностью загрузить при указанных выше условиях.

## Примеры

Стандартный ввод
5 7
Стандартный вывод
4

## Пример программы-решения

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 signed main(){
5     int a, b;
6     cin >> a >> b;
7     int b1 = min(a, b / 2);
8     a -= b1;
9     b -= b1 * 2;
10    cout << b1 + a / 2 << endl;
11 }
```

## Задача 4.2.1.2. Три города (15 баллов)

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

### Условие

Три города находятся на одной прямой: Первый и Третий с краев, Второй — где-то между ними. Между Первым и Третьим городом можно перемещаться либо на поезде, либо на самолете, при этом все перемещения проходят через Второй город. Более того, мэр Второго города, в целях поддержки обоих видов транспорта, обязал всех приехавших во Второй город поездом улетать из него на самолете и наоборот, всех прилетевших самолетом — уезжать из Второго города поездом.

Для определенности, пусть расстояние между Первым и Вторым городами равно  $A$ , расстояние между Вторым и Третьим городами равно  $B$ , причем  $A \leq B$ . Известно, что если из Первого во Второй лететь самолетом, а затем из Второго в Третий ехать поездом, время в пути составит  $X$  ч. Если же из Первого во Второй ехать поездом, а затем из Второго в Третий лететь самолетом, то время в пути составит  $Y$  ч.

Самолет летит со скоростью, в  $k$  раз превышающей скорость поезда. По заданным числам  $X$ ,  $Y$ ,  $k$  требуется определить, во сколько раз расстояние  $B$  больше расстояния  $A$ .

### **Формат входных данных**

В одной строке через пробел заданы три целых числа:

- $X$  — время движения, если из Первого во Второй город лететь самолетом, а затем из Второго в Третий ехать поездом;
  - $Y$  — время движения, если из Первого во Второй город ехать поездом, а затем из Второго в Третий лететь самолетом;
  - $k$  — коэффициент, показывающий, во сколько раз самолет быстрее поезда.
- $1 \leq X, Y \leq 7 \cdot 10^5$ ;  
 $2 \leq k \leq 1000$ ;  
 $X \geq Y$ .

### **Формат выходных данных**

Вывести одно число — отношение, показывающее, во сколько раз расстояние  $B$  больше расстояния  $A$ .

Тесты, на которых будет проверяться решение, сгенерированы таким образом, что ответ всегда является целым числом.

### **Примеры**

#### *Пример №1*

<b>Стандартный ввод</b>
5 3 3
<b>Стандартный вывод</b>
3

#### *Пример №2*

<b>Стандартный ввод</b>
12 12 3
<b>Стандартный вывод</b>
1

### **Примечания**

Приведем пример, показывающий, как получается ответ для первого теста из условия. Обращаем внимание, что это не единственный пример расстояний, для

которого возможен такой набор данных. В то же время, искомое отношение определяется входными данными всегда однозначно.

После некоторых рассуждений можно получить, что если, например, взять расстояние  $A$  равным 15 км, а расстояние  $B$  равным 45 км, скорость поезда равной 10 км/ч, а скорость самолета равной 30 км/ч, то получим требуемые результаты:

$$\frac{15}{30} + \frac{45}{10} = 5,$$

$$\frac{15}{10} + \frac{45}{30} = 3.$$

Отношение  $\frac{B}{A}$  равно  $\frac{45}{15}$  и равно 3.

### **Пример программы-решения**

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int X, Y, k;
5     cin >> X >> Y >> k;
6     int rat= (X * k - Y) / (Y * k - X);
7     cout << rat << endl;
8 }
```

### **Задача 4.2.1.3. Круглая география (20 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 128 Мбайт.

### **Условие**

Чего только ни встретишь в глубинах космоса! Есть, например, удивительная планета, покрытая океаном. В этом океане встречаются острова. Они имеют форму абсолютно правильного круга. Внутри этих островов могут быть озера. Они тоже имеют форму круга. Внутри этих озер могут быть свои острова. И верно, они тоже имеют форму круга. Внутри этих островов... Ну, вы поняли.

Итак, задано множество окружностей, описывающих эту круглую географию. Для каждой окружности требуется выяснить, что она собой представляет: остров (`Island`) или озеро (`Lake`).

### **Формат входных данных**

Будем считать, что вся планета исходно покрыта океаном, и только незначительную часть ее поверхности занимают острова. Представим эту область, содержащую острова, в виде прямоугольной поверхности (карты) и упрощенно будем считать, что эта поверхность является плоской. Вся остальная часть планеты, не входящая на карту, покрыта океаном, и не представляет для нас интереса.

В первой строке ввода задано одно натуральное число  $n$  — количество окружностей,  $1 \leq n \leq 1000$ . Далее в  $n$  строках заданы по три целых числа  $x, y, R$  через пробел — описание очередной окружности. Числа  $x$  и  $y$  задают координаты центра ( $-100 \leq x, y \leq 100$ ), а  $R$  — радиус окружности ( $1 \leq R \leq 100$ ).

Гарантируется, что никакие две окружности на входе не пересекаются и не касаются.

### **Формат выходных данных**

Вывести в ответ  $n$  строк. Для каждой окружности на входе в соответствующую строку вывести либо слово **Island**, если она является островом, либо **Lake**, если она является озером. Порядок вывода должен совпадать с порядком окружностей на входе.

### **Примечания**

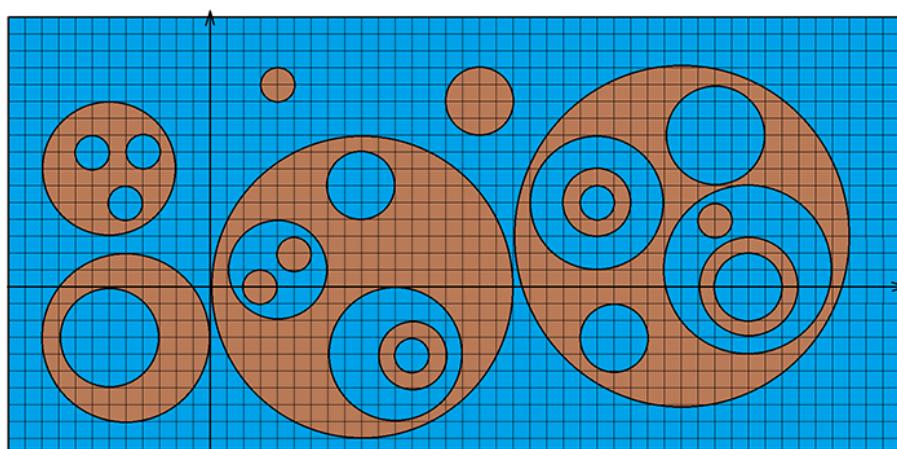


Рис. 4.2.1

На рис. 4.2.1 представлена карта для примера из условия. Синим обозначены водные пространства, коричневым — острова.

**Примеры****Стандартный ввод**

```
26
-6 7 4
-7 8 1
-5 5 1
-4 8 1
-5 -3 5
-6 -3 3
3 12 1
16 11 2
9 0 9
9 6 2
4 1 3
3 0 1
5 2 1
11 -4 4
12 -4 2
12 -4 1
28 3 10
23 5 4
23 5 2
23 5 1
24 -3 2
30 9 3
32 1 5
30 4 1
32 0 2
32 0 3
```

**Стандартный вывод**

```
Island
Lake
Lake
Lake
Island
Lake
Island
Island
Island
Lake
Lake
Island
Island
Lake
Island
Island
Lake
Island
Lake
Lake
Lake
Island
Lake
Island
Island
```

## Пример программы-решения

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 #define sz(a) (int)a.size()
3 #define pb push_back
4 #define all(a) a.begin(), a.end()
5 #define for0(i, n) for(int i = 0; i < n; i++)
6 #define for1(i, n) for(int i = 1; i <= n; i++)
7 #define x first
8 #define y second
9 #define int long long
10 using namespace std;
11 typedef pair<int, int> pii;
12 typedef vector<int> vi;
13 const int INF = 1e18;
14 const int MOD = 1e9 + 7;
15 const int LG = 19;

16
17
18 struct circ{
19     pair<double, double> c;
20     double R;
21 };
22
23
24 double dist(pair<double, double> a, pair<double, double> b){
25     return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y -
26     b.y));
27 }
28
29
30
31 bool inside(circ inc, circ outc){
32     return (dist(inc.c, outc.c) + inc.R < outc.R);
33 }
34
35
36 signed main(){
37     int n;
38     cin >> n;
39     vector<circ> v(n);
40     for(int i = 0; i < n; i++){
41         cin >> v[i].c.x >> v[i].c.y >> v[i].R;
42     }
43     vi mask(n, 0);
44     for(int i = 0; i < n; i++){
45         for(int j = 0; j < n; j++){
46             if(i != j && inside(v[i], v[j])){
47                 mask[i]++;
48             }
49         }
50     }
51     for(int i = 0; i < n; i++){
52         cout << ((mask[i] % 2)? "Lake\n" : "Island\n");
53     }
54 }
```

### **Задача 4.2.1.4. Мэллори подменяет сообщение (25 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 64 Мбайт.

#### **Условие**

Алиса посыпает Бобу сообщение, которое состоит из  $n$  положительных целых чисел. Активная злоумышленница Мэллори пытается нарушить целостность сообщения. Она перехватывает эти посыпаемые числа и по очереди заменяет их на другие, после чего посыпает эти искаженные числа Бобу.

Мэллори не может менять полученные числа произвольным образом. Ее алгоритм построения искаженных чисел следующий: исходно у нее было число  $t_0 = 0$ . Перехватив очередное число Алисы  $a_i$ , Мэллори берет имеющееся у себя на данный момент число  $t_{i-1}$  и формирует из него новое число  $t_i$  либо как  $t_i = a_i + t_{i-1}$  — этот вариант изменения доступен всегда, либо как  $t_i = a_i - t_{i-1}$ , такой вариант доступен, если  $a_i > t_{i-1}$ , то есть получающееся в итоге число является положительным и не противоречит характеру посыпаемых чисел. После этого Мэллори пересыпает полученное число  $t_i$  Бобу и использует новое  $t_i$  для следующей подмены.

Если вариант замены единственный ( $t_{i-1} \geq a_i$ ), то Мэллори делает замену  $t_i = a_i + t_{i-1}$  без вариантов. Если же возможны оба варианта замены, то Мэллори поступает случайным образом, и на выходе может получиться как  $t_i = a_i + t_{i-1}$ , так и  $t_i = a_i - t_{i-1}$ . При этом известно, что вариант подмены с вычитанием Мэллори сделает ровно  $k$  раз из  $n$ .

Получив вместо последовательности  $a_1, a_2, \dots, a_n$  последовательность  $t_1, t_2, \dots, t_n$ , и, зная все проделки Мэллори, Боб хочет понять объем работы по восстановлению правильного сообщения. По заданному числу  $k$  и последовательности  $t_i$  он просит вывести сумму всех возможных последовательностей, которые потенциально могла бы исходно выслать ему Алиса. Если нет ни одного сообщения, из которого могла бы получиться последовательность  $t_i$ , вывести 0.

Так как ответ может оказаться очень большим, нужно вывести его остаток от деления на число  $10^9 + 7$ .

#### **Формат входных данных**

В первой строке содержатся два целых числа  $n$  и  $k$  через пробел,  $1 \leq n \leq 1\,000$ ,  $0 \leq k \leq n$ .

Во второй строке содержатся  $n$  целых чисел  $t_i$  через пробел,  $1 \leq t_i \leq 10^5$ .

#### **Формат выходных данных**

Вывести одно число — сумму всех возможных последовательностей, которые могла бы потенциально послать Алиса таких, что применив к каждой из них неко-

торым образом свой алгоритм подмены, Мэллори получит последовательность  $t_i$ , причем вычитание в каждом из этих вариантов она произведет ровно  $k$  раз.

### **Примеры**

*Пример №1*

<b>Стандартный ввод</b>
5 3 2 5 8 7 12
<b>Стандартный вывод</b>
252

*Пример №2*

<b>Стандартный ввод</b>
5 0 2 5 8 7 12
<b>Стандартный вывод</b>
0

*Пример №3*

<b>Стандартный ввод</b>
5 1 2 5 8 7 12
<b>Стандартный вывод</b>
28

### **Примечания**

Рассмотрим первый пример:  $n = 5$ ,  $k = 3$ . Существует ровно шесть последовательностей, которые исходно могла бы подготовить Алиса для пересылки таких, что, применив к каждой из них некоторым образом алгоритм подмены, Мэллори получит 2, 5, 8, 7, 12. Приведем все эти последовательности:

- 2, 7, 3, 15, 5 ( $2 - 0 = 2$ ,  $7 - 2 = 5$ ,  $3 + 5 = 8$ ,  $15 - 8 = 7$ ,  $5 + 7 = 12$ );
- 2, 3, 13, 15, 5 ( $2 - 0 = 2$ ,  $3 + 2 = 5$ ,  $13 - 5 = 8$ ,  $15 - 8 = 7$ ,  $5 + 7 = 12$ );
- 2, 3, 3, 15, 19 ( $2 - 0 = 2$ ,  $3 + 2 = 5$ ,  $3 + 5 = 8$ ,  $15 - 8 = 7$ ,  $19 - 7 = 12$ );
- 2, 7, 3, 15, 19 ( $2 + 0 = 2$ ,  $7 - 2 = 5$ ,  $3 + 5 = 8$ ,  $15 - 8 = 7$ ,  $19 - 7 = 12$ );
- 2, 7, 13, 15, 5 ( $2 + 0 = 2$ ,  $7 - 2 = 5$ ,  $13 - 5 = 8$ ,  $15 - 8 = 7$ ,  $5 + 7 = 12$ );
- 2, 3, 13, 15, 19 ( $2 + 0 = 2$ ,  $3 + 2 = 5$ ,  $13 - 5 = 8$ ,  $15 - 8 = 7$ ,  $19 - 7 = 12$ ).

В каждом случае для достижения последовательности  $t_i$  в исходной последовательности Мэллори производит ровно три вычитания. Можно видеть, что сумма всех чисел во всех этих последовательностях равна 252.

Во втором примере  $k = 0$ . Можно показать, что Алиса не может сгенерировать ни одной последовательности, из которой можно получить заданную последовательность  $t_i$ , ни разу не использовав подмены с вычитанием. Таким образом, ответ равен 0.

### Пример программы-решения

Ниже представлено решение на языке C++.

C++

```

1 #include<bits/stdc++.h>
2 #define sz(a) (int)a.size()
3 #define pb push_back
4 #define all(a) a.begin(), a.end()
5 #define for0(i, n) for(int i = 0; i < n; i++)
6 #define for1(i, n) for(int i = 1; i <= n; i++)
7 #define x first
8 #define y second
9 #define int long long
10 using namespace std;
11 typedef pair<int, int> pii;
12 typedef vector<int> vi;
13 typedef vector<vector<int>> vvi;
14 const int INF = 1e18;
15 const int MOD = 1e9 + 7;
16 const int LG = 19;
17
18 signed main(){
19     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
20     int n, k;
21     cin >> n >> k;
22     vi v(n + 1, 0);
23     for1(i, n){
24         cin >> v[i];
25     }
26     vi add(n + 1, -1), subs(n + 1, 0);
27     for1(i, n){
28         if(v[i] > v[i - 1]){
29             add[i] = v[i] - v[i - 1];
30         }
31         subs[i] = v[i] + v[i - 1];
32     }
33     vector<vector<pii>> dp(n + 1, vector<pii>(n + 1, {0, 0}));
34     dp[0][0] = {0, 1};
35     for1(j, n){
36         for0(i, n + 1){
37             if(i > 0){
38                 dp[i][j].x = (dp[i - 1][j - 1].x + (dp[i - 1][j - 1].y *
39                     subs[j]) % MOD) % MOD;
40                 dp[i][j].y = dp[i - 1][j - 1].y;
41             }
42             if(add[j] > 0){
43                 (dp[i][j].x += dp[i][j - 1].x + (dp[i][j - 1].y * add[j])) %
44                     MOD) %= MOD;
45             }
46         }
47     }
48 }
```

```

43             (dp[i][j].y += dp[i][j - 1].y) %= MOD;
44         }
45     }
46 cout << dp[k][n].x << endl;
47 }
48 }
```

### **Задача 4.2.1.5. Обезьяна и Word (30 баллов)**

**Имя входного файла:** стандартный ввод или `input.txt`.

**Имя выходного файла:** стандартный вывод или `output.txt`.

**Ограничение по времени выполнения программы:** 1 с.

**Ограничение по памяти:** 128 Мбайт.

#### **Условие**

Одна обезьяна решила проверить известную гипотезу об обезьянах и печатных машинках. Так как на дворе был XXI век, обезьяна открыла Word и начала печатать в нем случайные слова.  $i$ -е по порядку слово она вставляла на  $p_i$ -ю позицию, после чего все ранее напечатанные слова, находящиеся в этой или более правой позиции, естественным образом сдвигались на длину этого слова правее. В итоге у обезьяны получился текст из одной строки, разбитой на слова, между которыми стояло ровно по одному пробелу. Знаков препинания, начальных и концевых пробелов в ее тексте не было. Все слова состояли из заглавных и прописных латинских букв.

По заданному порядку слов, которые печатала обезьяна, и позициям, в которые она их вставляла, нужно получить окончательный текст.

#### **Формат входных данных**

В первой строке находится число  $n$  — количество слов, которые напечатала обезьяна,  $1 \leq n \leq 10^5$ .

В следующих  $n$  строках находятся по одному слову  $w_i$  и одному числу  $pos_i$  через пробел. Это означает, что обезьяна вставила слово  $w_i$  так, что оно стало на позицию  $pos_i$ , при этом все ранее напечатанные слова, имеющие такую или большую позицию, сместились на длину этого слова и одного пробела вправо. Слово  $w_i$  состоит из заглавных и прописных латинских букв и имеет длину не более 9. Позиция  $pos_i$ , куда его вставила обезьяна, находится в пределах от 1 до  $i$ .

#### **Формат выходных данных**

Вывести в одну строку итоговый текст, полученный обезьянкой. Слова должны разделяться между собой ровно одним пробелом.

## Примеры

### Пример №1

#### Стандартный ввод

```
5
Tolstoy 1
Peace 2
and 2
Leo 1
War 3
```

#### Стандартный вывод

```
Leo Tolstoy War and Peace
```

### Пример №2

#### Стандартный ввод

```
7
a 1
bb 1
ccc 1
dddd 1
eeeeee 1
fffffff 1
gggggggg 1
```

#### Стандартный вывод

```
gggggggg ffffff eeeee dddd ccc bb a
```

## Пример программы-решения

Ниже представлено решение на языке C++.

### C++

```

1 #include<bits/stdc++.h>
2 #define sz(a) (int)a.size()
3 #define pb push_back
4 #define all(a) a.begin(), a.end()
5 #define for0(i, n) for(int i = 0; i < n; i++)
6 #define for1(i, n) for(int i = 1; i <= n; i++)
7 #define x first
8 #define y second
9 #define int long long
10 using namespace std;
11 typedef pair<int, int> pii;
12 typedef vector<int> vi;
13 const int INF = 1e18 + 1;
14 const int MOD = 1e9 + 7;
15 const int LG = 19;
16 const int N = (1 << LG);
```

```
17 vi tr(2 * N, 1);
18 vi res(N, 0);
19 void build(){
20     for(int i = N - 1; i >= 0; i--){
21         tr[i] = tr[2 * i] + tr[2 * i + 1];
22     }
23 }
24
25 void push(int t, int a, int cpos){
26     if(t >= N){
27         res[t - N] = a;
28         return;
29     }
30
31     if(cpos <= tr[2 * t]){
32         tr[2 * t]--;
33         push(2 * t, a, cpos);
34     }
35     else{
36         tr[2 * t + 1]--;
37         push(2 * t + 1, a, cpos - tr[2 * t]);
38     }
39 }
40
41 signed main(){
42     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
43     int n;
44     cin >> n;
45     vector<string> v(n);
46     vi pos(n);
47     for0(i, n){
48         cin >> v[i] >> pos[i];
49     }
50     build();
51     for(int i = n - 1; i >= 0; i--){
52         push(1, i, pos[i]);
53     }
54     for0(i, n - 1){
55         cout << v[res[i]] << ' ';
56     }
57     cout << v[res[n - 1]];
58     cout << endl;
59 }
```

## 4.2.2. Математика. 8–9 классы

### Задача 4.2.2.1. (15 баллов)

Тема: алгебра.

#### Условие

На сетке расставлены фишечки следующим образом: по одной фишке в центре каждой из ячеек и по одной фишке — в каждом из узлов сетки.

Для примера на рис. 4.2.2 приведена сетка размером  $2 \times 3$ , на которой 18 фишечек (фишечки отмечены синим цветом).

Найдите количество фишечек на сетке размером  $25 \times 24$  ячейки.

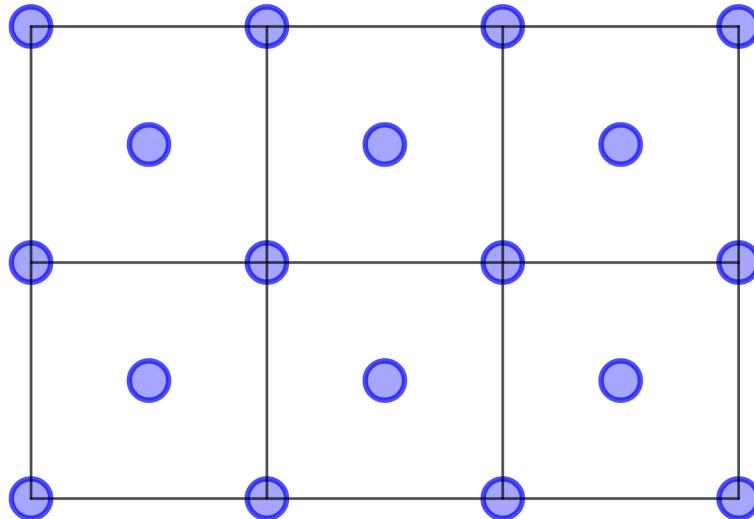


Рис. 4.2.2

#### Решение

Сетка содержит  $25 \times 24 = 600$  ячеек, таким образом, 600 фишечек находятся в центрах ячеек.

Количество фишечек в узлах  $(25 + 1) \cdot (24 + 1) = 650$ , так как количество горизонтальных и вертикальных линий, ограничивающих ячейки, на единицу больше, чем число ячеек в строке или столбце.

Значит, всего фишечек  $600 + 650 = 1\,250$ .

**Ответ:** 1 250.

### ***Критерии оценивания***

Только ответ без объяснения — 5 баллов.

Верно посчитано общее количество фишек в центрах ячеек — 5 баллов.

Верно посчитано общее количество фишек в узлах ячеек — 10 баллов.

Арифметическая ошибка при верной логике решения — 12 баллов.

Найдены 600 и 650, но не найдена сумма — баллы не снимаются.

### ***Задача 4.2.2.2. (20 баллов)***

*Темы: комбинаторика, игры и стратегии.*

#### ***Условие***

В стране 20 городов, некоторые из которых соединены дорогами. Каждая дорога начинается в одном городе, заканчивается в другом и не проходит через остальные города. Дороги не пересекаются. Автолюбители Петя и Коля играют в следующую игру. Вначале Петя определяет два города, в первый из которых перемещается сам, а во второй перемещается Коля. Далее передвигаются по очереди, начинает Коля. За один ход игрок выбирает дорогу, ведущую из города, в котором он находится и едет по ней, попадая в другой город. Если после очередного хода оба игрока оказываются в одном городе, то сделавший ход игрок выигрывает. При каком наибольшем числе дорог может оказаться так, что у Пети есть выигрышная стратегия?

#### ***Решение***

Оценка. Заметим, что наибольшее число дорог в стране будет в том случае, если каждый город соединен с каждым. В такой стране  $\frac{20 \cdot 19}{2} = 190$  дорог. Но если в стране 190 дорог, то Коля первым же своим ходом может приехать в город, в котором находится Петя (каждый город соединен с каждым). Следовательно, в этом случае у Пети нет выигрышной стратегии. Поэтому в стране меньше 190 дорог, т. е. не больше 189 дорог.

Пример. Рассмотрим страну, в которой соединены дорогами все пары городов, кроме пары  $(A, B)$ . В такой стране 189 дорог. Выигрышная стратегия Пети заключается в том, чтобы выбрать для себя город  $A$ , для Коли — город  $B$ . В таком случае Коля не сможет выиграть первым своим ходом, и в какой бы город  $D$  Коля ни поехал, Петя перейдет из  $A$  в  $D$  своим первым ходом и выиграет.

**Ответ:** 189.

### ***Критерии оценивания***

Получен только ответ без рассуждений или с неверными рассуждениями — 5 баллов (баллы за этот пункт не складываются с баллами за следующие пункты).

Доказана «оценка», что городов не более 189 — 10 баллов.

Построен «пример» в котором 189 дорог и приведена стратегия за Петю — 10 баллов.

Построен «пример» в котором 189 дорог, но нет стратегии за Петю — 0 баллов.

### **Задача 4.2.2.3. (20 баллов)**

*Тема: метод от противного.*

#### **Условие**

На пальме растут 55 бананов, которые нужно распределить между девятью обезьянами так, чтобы любые две обезьяны получили в сумме хотя бы 9 бананов (некоторые обезьяны могут получить 0 бананов). Докажите, что по крайней мере три обезьяны получат одинаковое количество бананов.

#### **Решение**

Предположим противное, что одинаковое количество бананов могут получить не более двух обезьян. Теперь рассмотрим два случая.

1. Каждая обезьяна получила хотя бы 5 бананов. Тогда по 5 бананов получили не более двух обезьян, по шесть бананов — не более двух обезьян и т. д. Поэтому все обезьяны получили не менее  $5 \cdot 2 + 6 \cdot 2 + 7 \cdot 2 + 8 \cdot 2 + 9 = 61$  банан. Противоречие.
2. Хотя бы одна обезьяна получила менее 5 бананов. Обозначим эту обезьяну за  $A$ , а за  $n$ ,  $n \leq 4$ , обозначим количество полученных ею бананов. Рассматривая пары обезьян, одна из которых  $A$ , получим, что каждая из оставшихся получила хотя бы  $9 - n$  бананов. Действуя аналогично предыдущему случаю, получим, что все обезьяны получили в сумме хотя бы  $n + (9 - n) \cdot 2 + (10 - n) \cdot 2 + (11 - n) \cdot 2 + (12 - n) \cdot 2 = 84 - 7n$  бананов. Учитывая, что  $n \leq 4$ , получим  $84 - 7n \geq 84 - 28 = 56$ . Противоречие.

#### **Критерии оценивания**

Рассмотрен лишь первый случай или доказано, что есть обезьяна получившая не более 4 бананов — 5 баллов.

Рассмотрен лишь второй случай — 15 баллов.

Второй случай делается перебором возможных значений  $n$ , т. е.  $n = 4, 3, 2, 1, 0$ . В этом переборе упущен или неверно рассмотрен один случай — 10 баллов за весь второй случай.

В этом переборе упущено более одного случая, при этом рассмотрен случай  $n = 4$  — 5 баллов за весь второй случай.

Рассмотрен лишь случай  $n = 4$  без попыток перебора — 5 баллов за весь второй случай.

В этом переборе упущено более одного случая, при этом не рассмотрен случай  $n = 4$  — 0 баллов за весь второй случай.

Баллы за первый и второй случаи складываются.

### **Задача 4.2.2.4. (20 баллов)**

Тема: геометрия.

#### **Условие**

Пусть  $\Omega$  — описанная окружность остроугольного треугольника  $ABC$ . Биссектрисы углов  $A$ ,  $B$  и  $C$  пересекают  $\Omega$  в точках  $A_1$ ,  $B_1$  и  $C_1$  соответственно, а биссектрисы углов  $A_1$ ,  $B_1$  и  $C_1$  треугольника  $A_1B_1C_1$  пересекают  $\Omega$  в точках  $A_2$ ,  $B_2$  и  $C_2$  соответственно. Известно, что наименьший угол треугольника  $ABC$  равен  $40^\circ$ . Найдите наименьший угол треугольника  $A_2B_2C_2$ .

#### **Решение**

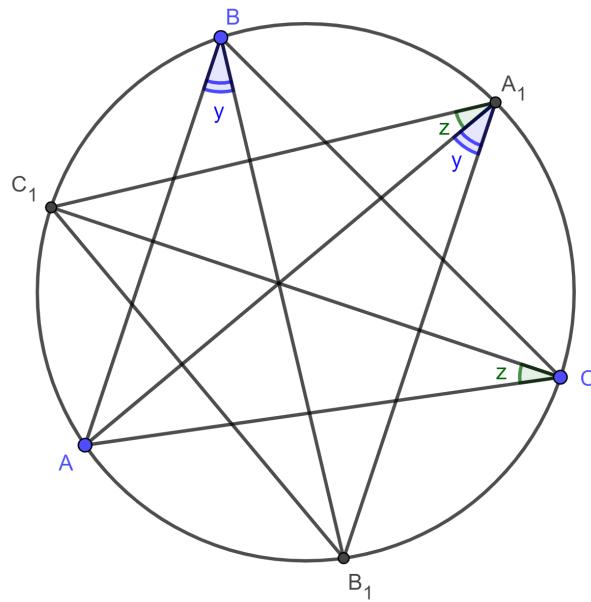


Рис. 4.2.3

Утверждение. Пусть дан треугольник  $ABC$  с углами  $2x$ ,  $2y$ ,  $2z$ , в котором его биссектрисы пересекают его описанную окружность в точках  $A_1, B_1, C_1$  соответственно. Тогда углы треугольника  $A_1B_1C_1$  равны  $x + y, y + z, z + x$ .

Доказательство. Заметим, что  $\angle C_1A_1A = \angle C_1CA$  как опирающиеся на дугу  $C_1A$  (см. рис. 4.2.3). При этом  $\angle C_1A_1A = \frac{1}{2}\angle ACB = z$ . Аналогично  $\angle B_1A_1A = y$ . Тогда  $\angle B_1A_1C_1 = y + z$ . Аналогично для других углов треугольника  $A_1B_1C_1$ . Утверждение доказано.

Обозначим теперь углы исходного треугольника  $4\alpha = 40^\circ, 4\beta, 4\gamma$  ( $\alpha \leq \beta \leq \gamma$ ). Применяя полученное выше утверждение для треугольника  $ABC$ , получим, что углы треугольника  $A_1B_1C_1$  равны  $\frac{4\alpha}{2} + \frac{4\beta}{2} = 2\alpha + 2\beta$ ,  $\frac{4\beta}{2} + \frac{4\gamma}{2} = 2\beta + 2\gamma$  и  $\frac{4\gamma}{2} + \frac{4\alpha}{2} = 2\gamma + 2\alpha$ .

Применяя теперь то же утверждение для треугольника  $A_1B_1C_1$  получим, что углы треугольника  $A_2B_2C_2$  равны  $\alpha + \beta + \beta + \gamma = \frac{4\alpha + 4\beta + 4\gamma}{4} + \beta = 45^\circ + \beta$ ,  $45^\circ + \gamma$ ,  $45^\circ + \alpha$ . Учитывая, что  $\alpha \leq \beta \leq \gamma$ , получим  $45^\circ + \alpha \leq 45^\circ + \beta \leq 45^\circ + \gamma$ . Поэтому наименьший угол треугольника  $A_2B_2C_2$  равен  $45^\circ + \alpha = 55^\circ$ .

**Ответ:**  $55^\circ$ .

### **Критерии оценивания**

Дан только верный ответ — 1 балл.

При верном решении используется утверждение без доказательства — снимается 2 балла.

Сформулировано утверждение без дальнейших продвижений — 3 балла.

Сформулировано и доказано утверждение без дальнейших продвижений — 5 баллов.

Вычислены углы треугольника  $A_2B_2C_2$  через углы треугольника  $ABC$ , ответ не получен — 10 баллов.

Вычислены углы треугольника  $A_2B_2C_2$  через углы треугольника  $ABC$  и посчитано значение одного из углов, равного  $55^\circ$ , но не доказано, что он наименьший — 15 баллов.

### **Задача 4.2.2.5. (25 баллов)**

*Тема: алгебра.*

#### **Условие**

Найдите всевозможные тройки целых чисел  $a, b, c$ , для которых выполняются равенства:  $a + b - c = 1$  и  $a^2 + b^2 - c^2 = -1$ .

#### **Решение**

Перепишем равенства в виде:

$$a + b = c + 1; \quad (4.2.1)$$

$$a^2 + b^2 = c^2 - 1. \quad (4.2.2)$$

Равенство (4.2.2) можно записать в виде  $(a+b)^2 - 2ab = c^2 - 1$ . Используя (4.2.1), получим  $(c+1)^2 - 2ab = c^2 - 1$  или  $c^2 + 2c + 1 - 2ab = c^2 - 1$ , если привести подобные и разделить обе части равенства на 2, получим:  $c + 1 = ab$ . Снова используя (4.2.1), получим  $a + b = ab \Leftrightarrow ab - a - b + 1 = 1 \Leftrightarrow (a-1)(b-1) = 1$ . Последнее уравнение имеет два решения в целых числах:

1.  $a - 1 = 1$  и  $b - 1 = 1$ , тогда получим тройку  $a = 2, b = 2, c = 3$ ;
2.  $a - 1 = -1$  и  $b - 1 = -1$ , тогда получим тройку  $a = 0, b = 0, c = -1$ .

**Ответ:**  $(2, 2, 3), (0, 0, -1)$ .

### **Критерии оценивания**

Найден только один ответ подбором и показано, что он подходит — 3 балла.

Найдены оба ответа подбором и показано, что они подходят — 5 баллов.

Получено равенство  $a + b = ab$  без дальнейших продвижений — 10 баллов.

Получено разложение  $(a - 1)(b - 1) = 1$  без дальнейших продвижений — 15 баллов.

Баллы за указанные выше продвижения и ответ (ответы) суммируются.

При решении уравнения в целых числах  $(a - 1)(b - 1) = 1$  потерян один из ответов, например, рассмотрен только случай  $a - 1 = 1, b - 1 = 1$  — 20 баллов.

Баллы за указанное выше продвижение и ответ (ответы) не суммируются.

## **4.2.3. Математика. 10–11 классы**

### **Задача 4.2.3.1. (15 баллов)**

*Темы: алгебра, тригонометрия.*

#### **Условие**

Сколько должно быть слагаемых под знаком корня в выражении

$$\sqrt[3]{\tan^3 \frac{\pi}{3} + \dots + \tan^3 \frac{\pi}{3}} = 6\sqrt{3},$$

чтобы равенство было верным?

#### **Решение**

Обозначим количество слагаемых под корнем через  $n$ , тогда  $\sqrt[3]{n \cdot \tan^3 \frac{\pi}{3}} = 6\sqrt{3}$ .

Вынося тангенс из-под корня и учитывая, что  $\tan \frac{\pi}{3} = \sqrt{3}$ , получим  $\sqrt[3]{n} = 6$ . Отсюда  $n = 6^3 = 216$ .

**Ответ:** 216.

### **Критерии оценивания**

Только ответ — 2 балла.

Записано уравнение, в котором неизвестным является количество слагаемых под корнем, например,  $\sqrt[3]{n \cdot \tan^3 \frac{\pi}{3}} = 6\sqrt{3}$  — 5 баллов.

Арифметическая ошибка при логически верном рассуждении — 10 баллов.

### **Задача 4.2.3.2. (15 баллов)**

*Тема: игры и стратегии.*

#### **Условие**

На доске выписаны все натуральные числа от 1 до 999 включительно. Миша и Ян ходят по очереди, начинает Миша. За один ход игрок стирает с доски одно из чисел. Игра заканчивается, когда на доске останется два числа. Если их сумма равна кубу какого-либо целого числа, то выигрывает Миша, в противном случае выигрывает Ян. Кто из игроков имеет выигрышную стратегию, позволяющую ему победить вне зависимости от ходов противника?

#### **Решение**

Опишем выигрышную стратегию Миши. Первым ходом Миша стирает число 500, а оставшиеся числа мысленно делит на следующие пары: (1, 999), (2, 998), ..., (499, 501). Далее, какое бы число ни вычеркнул Ян, своим очередным ходом Миша вычеркивает число из той же пары. После 997-го хода, который сделает Миша, останется два числа, причем оба будут входить в одну и ту же пару. Осталось заметить, что сумма чисел в каждой паре равна 1000, что является кубом целого числа.

**Ответ:** Миша имеет выигрышную стратегию.

#### **Критерии оценивания**

Верно указан Миша как игрок, имеющий выигрышную стратегию — 1 балл.

Показано, что последний ход делает Миша — 1 балл.

Приведено разбиение на 499 пар таких, что сумма чисел в каждой паре является кубом целого числа — 5 баллов.

Баллы, указанные в критериях выше, складываются. И не складываются с баллами в нижеследующих критериях.

Приведена выигрышная стратегия за Мишу, возможно, отличающаяся от стратегии, содержащейся в авторском решении, но не доказано, что стратегия выигрышная — 12 баллов.

### **Задача 4.2.3.3. (20 баллов)**

*Тема: алгебра и теория чисел.*

#### **Условие**

Пусть  $x_1, x_2, x_3, x_4, x_5$  — последовательные натуральные числа (именно в таком порядке) такие, что  $x_1 + x_2 + x_3 + x_4 + x_5$  — точный куб, а  $x_2 + x_3 + x_4$  — точный квадрат. Найдите наименьшее возможное значение, которое может принимать  $x_3$ .

***Решение***

Обозначим  $x_3$  через  $n$ . Тогда  $x_1 + x_2 + x_3 + x_4 + x_5 = 5n$  и  $x_2 + x_3 + x_4 = 3n$ .

Так как  $5n = m^3$  и 5 — простое число, то  $m$  делится на 5,  $m^3$  делится на  $5^3 = 125$ , а значит,  $n$  делится на 25.

С другой стороны,  $3n = k^2$ , значит,  $k$  делится на 3, значит,  $n$  делится на 3.

Так как  $(5, 3) = 1$  и  $5n = m^3$ , то  $n$  делится на 27. Таким образом,  $n$  делится на  $25 \cdot 27 = 675$ , поэтому  $n \geq 675$ .

Убедимся, что  $n = 675$  подходит:  $673 + 674 + 675 + 676 + 677 = (3 \cdot 5)^3 = 15^3$ ,  $674 + 675 + 676 = 2025 = 45^2$ .

**Ответ:** 675.

***Критерии оценивания***

Только ответ без объяснений — 0 баллов.

Ответ и проверка, что ответ подходит — 2 балла.

Доказано, что  $x_3$  делится на 3 — 3 балла.

Доказано, что  $x_3$  делится на 5 — 3 балла (суммируется с предыдущим).

Доказано, что  $x_3$  делится на 27 — 6 баллов.

Доказано, что  $x_3$  делится на 25 — 6 баллов (суммируется с предыдущим).

Доказано, что  $x_3 \geq 675$  — 15 баллов.

***Задача 4.2.3.4. (25 баллов)***

*Тема: геометрия.*

***Условие***

Дана окружность  $\omega$  с диаметром  $AB$ . На окружности выбрана точка  $N$ , отличная от точек  $A$  и  $B$ . Перпендикуляр из  $N$  к  $AB$  пересекает  $AB$  в точке  $M$  и повторно пересекает  $\omega$  в точке  $K$ . Окружность с центром  $N$  и радиусом  $NM$  пересекает  $\omega$  в точках  $P_1$  и  $P_2$ . Точка  $L$  — точка пересечения  $P_1P_2$  с отрезком  $KN$ . Докажите, что  $KP_1$  вдвое больше  $P_1L$ .

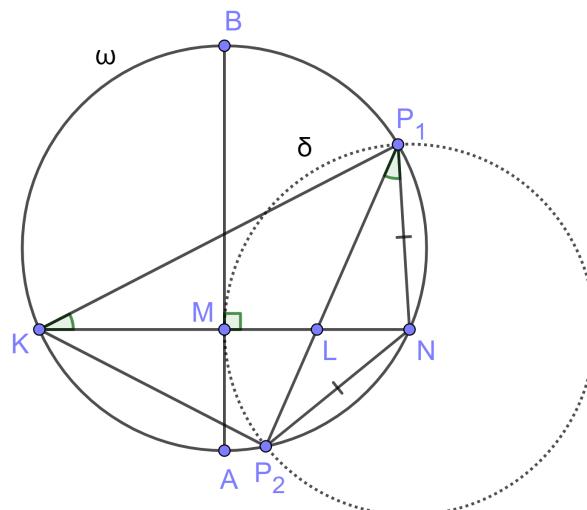
**Решение**

Рис. 4.2.4

Обозначим окружность с центром  $N$  и радиуса  $NM$  за  $\delta$ .

Так как хорды  $P_1N$  и  $P_2N$  равны как радиусы окружности  $\delta$ , то равны и дуги  $P_1N$  и  $P_2N$ . Следовательно, равны углы  $\angle NKP_1 = \angle NP_1P_2$  как опирающиеся на равные дуги. Тогда треугольники  $NKP_1$  и  $NP_1L$  подобны по двум углам ( $\angle NKP_1 = \angle NP_1L$  и  $\angle KNP_1$  — общий). Тогда  $\frac{KP_1}{P_1L} = \frac{NK}{NP_1}$ . Заметим теперь, что  $NP_1 = NM$  как радиусы окружности  $\delta$  и  $NM = MK$ , так как диаметр делит перпендикулярную ему хорду пополам. Поэтому  $\frac{NK}{NP_1} = \frac{2NM}{NM} = 2$ , откуда  $\frac{KP_1}{P_1L} = 2$ .

**Критерии оценивания**

Упоминается без доказательства, что диаметр делит перпендикулярную хорду пополам — баллы не снимаются.

Доказано, что треугольники  $NKP_1$  и  $NP_1L$  подобны — 10 баллов.

**Задача 4.2.3.5. (25 баллов)**

Темы: теория чисел, комбинаторика.

**Условие**

Дано натуральное  $n = p^k$ , где  $p$  — простое, а  $k$  — нечетное. На доску выписали все натуральные делители  $n$  (в том числе 1 и само  $n$ ). Юра разбил выписанные числа на пары, в каждой паре посчитал произведение чисел и все эти произведения выписал в тетрадку. Оказалось, что все числа в тетрадке имеют одинаковое количество натуральных делителей. Докажите, что у Юры есть ровно один способ такого разбиения делителей числа  $n$  на пары.

### **Решение**

Обозначим  $d(x)$  — количество натуральных делителей числа  $x$ .

*Лемма.* Количество делителей числа  $p^m$ , где  $p$  — простое, равно  $m+1$ .

Доказательство. Делители  $p^m$  — это числа  $1, p, p^2, \dots, p^m$ . Их ровно  $m+1$ . Лемма доказана.

У числа  $n = p^k$  следующие делители:  $1, p, p^2, \dots, p^k$ .

Рассмотрим какой-нибудь способ разбиения Юрай делителей числа  $n$  на пары. Пусть  $p^x$  — делитель, попавший в пару с  $1$ , а  $p^y$  — делитель, попавший в пару с  $p^k$ . Тогда  $d(1 \cdot p^x) = d(p^y \cdot p^k)$ . Откуда  $d(p^x) = d(p^{y+k})$ , значит,  $x+1 = y+k+1$ . Учитывая, что  $x \leq k$  и  $y \geq 0$ , получим  $x = k$ ,  $y = 0$ . Т. е.  $1$  и  $p^k$  попали в одну пару. Следовательно, число делителей в каждой паре равно  $d(1 \cdot p^k) = k+1$ .

Рассмотрим делитель  $p^m$  числа  $n$ . К нему в пару попадет такой делитель  $p^t$ , что  $d(p^m \cdot p^t) = k+1$  или  $m+t+1 = k+1$ . Откуда  $t = k-m$ . Учитывая, что  $k$  нечетное, получим, что  $t$  и  $m$  разной четности, и  $p^m \neq p^t$ , а значит,  $p^t$  определяется однозначно.

Разбиение всех делителей на пары вида  $\left(d, \frac{n}{d}\right)$  удовлетворяет условию задачи, так как в этом случае равны все произведения чисел в парах, а значит, равны и их количества делителей.

### **Ответ:**

#### **Критерии оценивания**

Приводится разбиение на пары, удовлетворяющие условию задачи — 5 баллов.

Доказана лемма — 2 балла.

Доказано, что делитель  $1$  должно быть в паре с числом  $n$  — 6 баллов.

Доказано, что если пару составляют делители  $p^m$  и  $p^t$ , то  $m+t = k - 12$  баллов.

Баллы за указанные выше критерии складываются.

В верном решении не доказывается, что если пару составляют делители  $p^m$  и  $p^t$ , то  $p^m \neq p^t$  — снять 5 баллов.

В решении не приводится разбиение на пары, удовлетворяющие условию задачи — снять 5 баллов.

## **4.3. Инженерный тур**

### **4.3.1. Общая информация**

В ходе заключительного этапа участники создают игровой прототип согласно техническому заданию и гайдлайну<sup>1</sup>. Результаты работы должны соответствовать рыночным критериям игровой индустрии и, при дальнейшем развитии, потенциально смогут стать коммерциализируемым продуктом.

В игровой индустрии России наблюдаются следующие ключевые проблемы, которые участники НТО решают в процессе разработки игр:

- Недостаток образовательных и полезных игр на российском рынке: в России существует ограниченное количество игр, которые совмещают развлечение с обучением, развитием когнитивных навыков или профессиональной ориентацией, поэтому от участников заключительного этапа требуется создание таких прототипов, которые смогут закрыть эту нишу, делая игры не только интересными, но и полезными.
- Ориентация на отечественный рынок: развитие российских игр, с учетом локального культурного контекста и запросов общества, остается актуальной задачей, и потому участники НТО работают над проектами, которые популяризируют отечественные ценности и продвигают российскую технологическую независимость, учитывая специфику российского рынка.

### **4.3.2. Легенда задачи**

Поздравляем! Вы прошли долгий путь, чтобы оказаться здесь, в самом центре главного события профиля Разработка компьютерных игр Национальной технологической олимпиады. Сегодня перед вами стоит задача, достойная настоящих разработчиков!

К вашей команде обращается представитель крупного игрового издателя с заманчивым предложением. Они готовы выпустить вашу игру под брендом Берлога — уникальной игровой вселенной, которая уже завоевала любовь тысяч игроков. Но чтобы это произошло, ваша игра должна стать частью этого мира и засиять на новом уровне!

Теперь все в ваших руках:

- Нужно изучить гайдлайны и сделать так, чтобы игра идеально им соответствовала. Проявите максимум творчества, чтобы создать что-то действительно захватывающее и оригинальное.
- Чтобы сделать игру еще более увлекательной, вам предлагают добавить одну из двух новых игровых механик. Выбор за вами, но помните: каждая из них должна раскрыть потенциал игры и по-настоящему увлечь будущих игроков.

---

<sup>1</sup>Гайдлайн (guideline) — это документ, в котором содержится набор правил по использованию цветов, логотипа, слогана, товарного знака, шрифтов и других составляющих фирменного стиля.

- Доведите демоверсию до совершенства, чтобы она соответствовала высоким стандартам индустрии.

Что на кону?

Если ваша команда справится лучше всех, ее ждет не только признание, но и возможность получить профессиональную поддержку от опытных специалистов и инвестиции в продвижение вашей игры.

Вызов принят?

Соберите свои силы, максимально зарядите фантазию и покажите миру, что именно ваша команда готова творить историю игровой индустрии!

### **4.3.3. Требования к команде и компетенциям участников**

Количество участников в команде: 3–4 человека.

Компетенции, которыми должны обладать члены команды:

1. **Геймдизайнер:** разрабатывает дизайн-документ проекта, структуру и правила игры; продумывает механики, игровой баланс, сюжет, персонажей и их взаимодействие с окружающим миром.
2. **Программист:** занимается созданием архитектуры проекта, реализацией взаимодействия объектов и версткой интерфейсов.
3. **Художник:** от стиля игры будет зависеть компетенция художника, которая необходима:
  - 2D-художник — создание 2D-персонажей, окружения, UX/UI;
  - 3D-художник — создание 3D-моделей персонажей, окружения.
4. **Тимлид:** руководит командой и разработкой, объединяет наработки в конечный продукт и демонстрирует проект.

### **4.3.4. Оборудование и программное обеспечение**

Таблица 4.3.1

Наименование	Описание
ПК или ноутбук с характеристиками не ниже следующих: <ul style="list-style-type: none"> <li>• операционная система Windows;</li> <li>• видеокарта gtx 1060 или мощнее, совместимая с DirectX 11 и поддержкой Shader Model 2,0 или выше;</li> <li>• процессор с тактовой частотой 3,0 ГГц или выше;</li> <li>• минимум 16 Гб оперативной памяти;</li> <li>• более 200 Гб свободного места на SSD.</li> </ul>	Позволяет запускать все необходимые программы — игровые движки, 3D- и графические редакторы, видеоредакторы и текстовые инструменты. Обеспечивает достаточную мощность для работы с проектами, программирования, рендеринга и тестирования игр.

Таблица 4.3.1

Наименование	Описание
Unity	Используется для создания логики игры, построения уровней, настройки физики, анимаций, интерфейсов (UI). Хорошо подходит для 2D и 3D-проектов.
Unreal Engine	Применяется для создания игр с реалистичной графикой. Предоставляет мощные инструменты визуального программирования (Blueprints), кинематографические эффекты, продвинутую физику и освещение.
Blender	Бесплатный софт для 3D-моделирования, текстурирования, анимации и рендеринга. Отлично подходит для инди-разработчиков.
3ds Max	Используется для создания 3D-объектов, окружений и визуальных эффектов.
Adobe Photoshop	Подходит для рисования концепт-артов, создания текстур, UI-элементов и иллюстраций.
Adobe Premiere Pro	Используется для монтажа трейлеров, кат-сцен, добавления титров и визуальных эффектов
Microsoft Word/Google Docs	Применяются для написания сценариев, диалогов, геймдизайн-документации (GDD) и заметок.
Notion	Удобен для организации задач, ведения документации, заметок и общего планирования проекта.

#### 4.3.5. Описание задачи

Необходимо выполнить 5 модулей заданий:

- Арт.
- Геймдизайн.
- Программирование.
- Продукт.
- Сеттинг.

Задание №1 — обязательное в блоках Геймдизайн, Арт, Программирование.

Задание №2 или №3 на выбор:

- если в Геймдизайне/Арте/Программировании выбран второй тип задания, то во всех остальных блоках тоже должен быть выполнен второй тип.
- если в Геймдизайне/Арте/Программировании выбран третий тип задания, то во всех остальных блоках тоже должен быть выполнен третий тип.

В блоке Продукт и Сеттинг все задания обязательны.

Задания по модулям:

- Геймдизайн (max 100 баллов):
  - ◊ Соответствие гайдлайну — max 50 баллов.
  - ◊ Процедурная генерация — max 50 баллов.

- ◊ Глобальные ивенты — max 50 баллов.
- Арт (max 100 баллов):
  - ◊ Соответствие гайдлайну — max 50 баллов.
  - ◊ Процедурная генерация — max 50 баллов.
  - ◊ Глобальные ивенты — max 50 баллов.
- Программирование (max 100 баллов):
  - ◊ Пользовательский контент — max 50 баллов.
  - ◊ Процедурная генерация — max 50 баллов.
  - ◊ Глобальные ивенты — max 50 баллов.
- Продукт (max 150 баллов):
  - ◊ Игровой прототип — max 60 баллов.
  - ◊ Презентация и питч — max 40 баллов (не идет в индивидуальный зачет, только в командный).
  - ◊ Страница в магазине — max 20 баллов.
  - ◊ Трейлер — max 30 баллов.
- Сеттинг и обучаемость (max 50 баллов):
  - ◊ Игровой тьюториал — max 25 баллов.
  - ◊ Интро — max 25 баллов.

### **Задача 4.3.5.1. Блок Геймдизайн (100 баллов)**

#### ***Соответствие гайдлайну (обязательное задание)***

##### ***Вступление***

При разработке игры важно держать ее фокус, чтобы не погрузиться в процесс добавления новых механик. Умение видеть конечный продукт — один из самых важных навыков геймдизайнера. Поэтому обязательное задание направлено на правку игры, чтобы сделать ее целостной, а не только набором механик. Именно этим и предстоит заниматься.

##### **1. Определение источников фана**

Проанализируйте популярную базовую статью Fourteen Forms of Fun <https://www.gamedeveloper.com/design/fourteen-forms-of-fun> (здесь предоставлен также машинный перевод этой статьи <https://disk.yandex.ru/i/5omdzoq0zCuU3w>). Напишите, каким двум формам фана должна в первую очередь соответствовать ваша игра. Для каждой из них ответьте на вопросы:

- Каков вид получения фана?
- Как достигается получение этого фана (общие слова)?
- Какие уже реализованные механики работают на получение этого вида веселья (конкретные механики и их аспекты)?

##### **2. Фокус игры**

Игры создаются для того, чтобы передать определенный опыт. В условиях ограниченности ресурсов лучше сделать упор на несколько ключевых механик

и развивать их, а не распылять небольшой бюджет на множество гиммиков, которые дадут спорный опыт, поэтому:

- определите три–четыре ключевые геймплейные механики (КГМ), опирающиеся на источники фана;
- для каждой укажите, какие механики дополняют КГМ (с небольшим описанием).

### 3. (не)Любимый фичекат

После выбора ключевых столпов игры, обозначения механик, которые их укрепляют, остается еще много механик, которые не работают (что не всегда очевидно) — они существуют в любой из игр. Найдите их, обоснуйте что они лишние и... вырежьте. От этого игра во многом станет лучше. В качестве альтернативы можно предложить изменение этих механик, чтобы они работали на ранее выбранные виды удовольствия и КГМ.

Прикрепите видео с изменением геймплея (до и после внесения поправок), в котором должно четко просматриваться, каким фрагмент геймплея был до переработки и каким он стал после этого, как преобразовалась механика (геймплей) в связи ее удалением. Длительность видеоролика — не более 3 мин.

Определите и измените/удалите как минимум две лишние механики для получения баллов за фичекат.

#### **Общий формат сдачи:**

- геймдизайн-документ (до 7 страниц, мин. шрифт — 12, Arial/Times New Roman);
- видео (строго до 3 мин), демонстрирующее работу.

## ***Формирование процедурной генерации в игре (задание на выбор)***

### ***Вступление***

В последние годы популярность игр с процедурной генерацией стремительно растет. Такие жанры, как «рогалик» стали невероятно востребованными благодаря своей высокой реиграбельности, уникальности каждого прохождения и глубине механик. Игроки ценят ощущение новизны в каждом забеге, а разработчики получают возможность создавать огромные миры без необходимости вручную проектировать каждую карту.

Процедурная генерация — это магия, которая делает каждое прохождение уникальным. Хотите бесконечный мир, как в Minecraft? Или случайные подземелья, как в Hades? Все это достигается с помощью комплексных алгоритмов, которые собирают весь контент в полноценные локации, уровни, не заставляя дизайнеров продумывать и строить множество уникального контента вручную.

Но в любой магии есть свои правила. Процедурная генерация может как обогатить геймплей, так и превратить его в хаос, где боссы спавнятся на стартовых локациях, а сундуки с легендарным оружием лежат под каждым кустом. Поэтому главное — не просто написать функцию random(), а сделать ее управляемой, честной и интересной!

В этом задании предстоит:

1. разработать систему процедурной генерации;
2. описать ее механики, влияние на геймплей, способы контроля случайности;
3. продемонстрировать в коротком видеоформате.

### **Задание**

#### **1. Обоснование необходимости процедурной генерации в игре**

Задачи:

- Описать три конкретные проблемы, которые решает процедурная генерация.
- Указать три аспекта влияния генерации на геймплей (реиграбельность, сложность, разнообразие).
- Объяснить два способа, с помощью которых случайная генерация вписывается в лор и атмосферу игры.

Процедурная генерация — это не просто способ создать больше контента, но и инструмент, который позволяет улучшить игровой процесс. Важно рассмотреть, как именно случайность решает реальные проблемы игры, а не только делает ее непредсказуемой или решает выдуманные проблемы.

#### **2. Элементы процедурной генерации**

Задачи:

- Указать не менее трех конкретных игровых элементов генерации в проекте (уровни, контент, параметры персонажей, комбинированный подход).
- Описать механику генерации для каждого элемента, которая генерирует не менее десяти уникальных единиц контента.
- Привести как минимум три примера уникальных игровых единиц из собственной игры для каждого элемента.

Приведенные элементы должны соответствовать основным игровым механикам, не противоречить игровой концепции и вписываться в лор и сеттинг игры. Кроме того, нужно показать, что процедурная генерация действительно создает качественное разнообразие, а не просто меняет цвета или числа.

#### **3. Контроль случайности**

Задачи:

- Описать три механики балансировки случайности (например, фиксированные шаблоны, гарантированные награды, «волновые» алгоритмы).
- Указать два правила генерации (например, минимальное и максимальное количество врагов, ограничения на комбинации предметов).
- Объяснить два метода для предотвращения создания слишком простых или сложных сценариев.

Механики балансировки — это базовые столпы генерации, с помощью которых игра не превращается в хаос, а стабильно держит предполагаемый геймдизайнером уровень сложности.

Правила генерации — это рычаги и цифры, с помощью которых геймдизайнер пытается достичь эталонного баланса.

Методы — здесь это дополнительные правила генерации, которые призваны сохранять кривую уровней сложности сценариев/локаций/прохождения для игрока. Они могут быть разными, но должны дополнять и соответствовать правилам генерации, описанным выше.

Важно объяснить, как игроку не дают случайно получить либо слишком легкую либо невозможную ситуацию. Подумайте, какие механики помогут сохранить баланс и сделать игру интересной.

#### **4. Взаимодействие игрока с процедурной генерацией**

Задачи:

- Описать три способа того, как игрок может влиять на генерацию (например, перевыборы, выбор параметров).
- Объяснить два тактических преимущества процедурной генерации для игрока.
- Описать две механики, позволяющие избежать фruстрации от неудачного рандома (например, система компенсации, выбор стартовых условий и другие).

Игрок не должен быть пассивным наблюдателем случайности. Важно продумать, как он может управлять процессом, минимизировать неудачи и использовать генерацию в свою пользу. Это создает стратегический элемент в игре.

#### **5. Описание первой части игры и видеодемонстрация процедурной генерации**

Задачи:

- Описать как минимум три ключевые моменты игрового опыта, которые получает игрок при прохождении первой части, задействуя все ранее описанные (см. п. 2 «Элементы процедурной генерации») элементы сгенерированного контента (например, выборы способностей, задания).
- Продемонстрировать в видео не менее трех различных элементов сгенерированного контента по пять единиц на каждый элемент (например, разные карты + комбинации предметов + вариации расположения врагов).
- Интегрировать в игру и продемонстрировать на видео необходимо строго ранее описанных в п. 2 элементов процедурной генерации.

Обратите внимание:

- ◊ Визуально должно быть понятно, как именно работает случайная генерация (например, с помощью коротких пояснительных вставок, динамики изменений, может быть добавлен закадровый комментарий или текстовые пояснения).
- ◊ Длительность видео строго до 3 мин.

Видео должно не просто демонстрировать рандомные события, а показывать, как генерация влияет на игровой процесс со стороны игрока. Подумайте, как сделать демонстрацию наглядной: можно использовать короткие текстовые пояснения, стрелки, сравнительные кадры, таймлапсы, разные билды или, например, построение тактики в зависимости от разного ландшафта.

### **Общий формат сдачи:**

- геймдизайн-документ (до 7 страниц, мин. шрифт — 12, Arial/Times New Roman);
- видео (строго до 3 мин), демонстрирующее работу процедурной генерации.

## ***Глобальные ивенты (задание на выбор)***

### ***Вступление***

Сегодня на рынок выходит все больше игр-сервисов. Представьте, что к вам пришел издатель и готов поддержать вас на солидную сумму, которой хватит и на всю разработку, и на пару вилл в Сочи. Но условия жесткие: за два дня нужно доказать, что ваш проект — это игра-сервис с большим потенциалом для развития, внедрив такие механики удержания, что AAA-играм не снились!

Перед вами огромный выбор. Это могут быть не просто сезонные ивенты от праздника к празднику, но и что-то более крупное! Все же играли (или хотя бы знают) про Fortnite, Genshin Impact, Concord, Clash of Clans и еще множество игр-сервисов. Так что выбор из того, что можно модифицировать огромный, а лучше придумать что-то совершенно оригинальное на основе существующих систем!

### ***Задание***

Необходимо выбрать два вида временных ивентов и описать их. Для того чтобы присутствовала альтернатива, нужно найти три системы и сделать обоснованный выбор.

#### **1. Анализ рынка**

Опишите три вида ивентов, которые могут хорошо подойти игре. При описании каждого опирайтесь на следующие пункты:

1. Общая формулировка. Опишите кратко и в общих словах этот вид глобальных ивентов.
2. Приведите три игры-референса с таким видом событий (игры-референсы этой механики не обязательно будут референсами к самому проекту):
  - объясните, почему ваша ЦА примет подобную механику (в этом поможет ответить схожесть вашей ЦА и ЦА референсов);
  - выберите две из трех игр; декомпозирийте реализацию ивентов в ней на механики с точки зрения user story (не нужно постоянно использовать одну и ту же формулировку — только пункты);
  - ответьте на вопрос: «Почему игроки участвуют и ждут эти события?».
3. Ответьте на вопросы: «Почему моей игре подходит этот вид временных событий?», «Как это временное событие поможет увеличить retention в моей игре?».

#### **2. Выбор двух систем**

Здесь все просто. Максимально обоснованно напишите, почему одна из трех идей хуже остальных (благодаря предыдущим пунктам вы определили, что

все они хорошо подходят вашему концепту, однако, постарайтесь найти узкие места в каждой).

### 3. Принцип работы

Опишите выбранные две системы по плану:

1. Опишите системы событий:

- охарактеризуйте контент, который представляет событие и то, какие эмоции, опыт он дает игроку (рекомендуется оформить в виде таблицы (не ссылки, а именно таблицы в данном документе!));
- приведите все способы получения контента игровым путем (при необходимости используйте таблицу для читаемости);
- укажите отличия от игр-референсов и то, какие решения выделяют вашу механику на фоне других.

2. Приведите конкретный пример наполнения события контентом из п. 3.1.

3. Укажите, как именно влияет получаемый контент на игровой процесс (пример: ускорение добычи ресурса X вдвое за недельный бонус).

4. Прикрепите блок-схему взаимодействия игрока с событием.

### 4. Покажите, что получилось

Прикрепите ссылку на видео, в котором показывается ваша реализация систем в проекте (продолжительность видео строго до 2 мин, ускорения более, чем в 2 раза на содержательной части не приветствуются).

### 5. Игра вдолгую

Опишите планы для итоговых систем:

1. Регулярность событий: как часто они проходят, какой промежуток между ними.
2. Представьте дорожную карту этих событий на 6–12 месяцев вперед (лучше изобразить в виде таймлайна):
  - отобразите даты начала событий;
  - укажите тематику или то, к чему привязано событие;
  - опишите особенности в механиках этого события.

#### Общий формат сдачи:

- геймдизайн-документ (до 7 страниц, мин. шрифт — 12, Arial/Times New Roman);
- видео (строго до 3 мин), демонстрирующее работу глобальных ивентов.

#### Правила оформление ответа на задания

Весь ответ — два pdf-файла и два видео. В них можно добавлять ссылки на гугл-таблицы и гугл-диск, в которых хранится часть ответа, требующая дополнительного описания в виде таблиц, картинок. Каждая ссылка из документа должна быть подписана и хранить только те объекты, которые представляют собой ответ на данный пункт задания, то есть для каждого ответа — отдельная ссылка.

## ***Решение***

Чтобы решить задачи, необходимо сначала сфокусироваться на формировании целостного игрового опыта. Это начинается с определения источников фана: нужно проанализировать формы удовольствия по статье Fourteen Forms of Fun и соотнести их с тем, какие эмоции и действия вызывает игра. Это позволяет избежать механической сборки фич и сосредоточиться на том, чтобы весь проект вызывал конкретное чувство у игрока — например, удовольствие от решения задач или открытия нового.

Далее, исходя из выбранных видов фана, выделяются ключевые геймплейные механики — те, что действительно создают игру и ее основное впечатление. Нужно внимательно оценить все фичи и определить, какие из них работают на эти механики, а какие — нет.

После этого можно переходить к фичекату: проанализировать второстепенные системы и убрать или переосмыслить как минимум две, которые либо не усиливают КГМ, либо дублируют другие механики, либо не дают игроку ценного опыта.

Важно также не только убрать механику, но и объяснить, почему без нее игра стала лучше, либо предложить, как ее улучшить, чтобы она работала на выбранные источники фана. Результатом работы станет документ с краткой, но логически выстроенной игровой моделью, и видео, демонстрирующее улучшение геймплея.

## ***Соответствие гайдлайну (обязательное задание)***

Ссылки на документы: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133004-CW21IaFFUh.pdf>, <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133644-IdlK55LSGo.pdf>.

Ссылки на видео: [https://disk.yandex.ru/d/QgndHaG0Ohofeg/GAMEDESIGN\\_1.mp4](https://disk.yandex.ru/d/QgndHaG0Ohofeg/GAMEDESIGN_1.mp4), [https://disk.yandex.ru/i/E02kiMLn1Mdm\\_g](https://disk.yandex.ru/i/E02kiMLn1Mdm_g).

## ***Формирование процедурной генерации в игре (задание на выбор)***

Ссылка на документ: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133640-XErQ28oELm.pdf>

Ссылка на видео: [https://disk.yandex.ru/i/RZ\\_cXkLmVHYK0g](https://disk.yandex.ru/i/RZ_cXkLmVHYK0g).

## ***Глобальные события (задание на выбор)***

Ссылка на документ: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133028-C1OmlhSeD8.pdf>.

Ссылка на видео: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/EVENTS.mp4>.

## **Задача 4.3.5.2. Блок Арт (100 баллов)**

### **Соответствие гайдлайну (обязательное задание)**

Требуется адаптировать проект под условия гайдлайна, в котором представлен перечень требований к нему. Это реальные критерии, которые могут выдвигать издатели и цифровые площадки. Они направлены, в первую очередь, на максимальное улучшение визуального качества проекта без создания нового контента.

Гайдлайн включает 11 разделов, каждый из которых содержит правила, применимые к любому проекту (игнорировать их нельзя). При выполнении может понадобиться не только улучшить, но и добавить/удалить/упростить существующие элементы.

В итоге соблюдение всех правил гайдлайна не только принесет команде баллы, но и повысит общую презентабельность проекта, что положительно скажется на оценке другими экспертами.

Гайдлайн: <https://cheerful-bearberry-f02.notion.site/1b9f25e0f9158086921ee468afc2dcde>.

#### **Общий формат сдачи:**

- видео (строго до 3 мин), демонстрирующее все меню и игровой процесс с игровым аудио, акцент на визуал, без комментариев;
- билд.

### **Процедурная генерация (задание на выбор)**

Вам следует рассчитать свои силы и время, чтобы создать большое количество разнообразных ассетов за короткое время, а также добиться высокого качества итогового внешнего вида сгенерированного контента.

Оценивается количество:

- новых уникальных ассетов, созданных для генерации (пример: ель, сосна, береза);
- категорий объектов, используемых для генерации (пример: деревья, камни, кусты);
- игровых элементов, созданных путем процедурной генерации (пример: локация, персонаж, карта).

Общее качество ассетов и сгенерированного контента также будет оцениваться.

#### **Общий формат сдачи:**

- pdf-файл (таблица, содержащая порядковый номер, пронумерованный игровой элемент в генерации которого объект участвует, пронумерованная категория объекта, изображение);
- видео (строго до 2 мин), демонстрирующее варианты контента, сгенерированного внутри игры;
- билд.

Таблица 4.3.2. Пример правильно оформленной таблицы

Номер	Элемент	Категория	Изображение
1	1. Локация	1. Дерево	(скриншот)
2	1. Локация	1. Дерево	(скриншот)
3	2. Персонаж	2. Плащ	(скриншот)
4	2. Персонаж	3. Шлем	(скриншот)

## ***Глобальные ивенты (задание на выбор)***

Вам предстоит сделать каждый ивент визуально заметным и выразительным, используя все возможные художественно-выразительные средства. Пользователь должен четко видеть, что ивент начался, что он все еще активен и что он завершился.

Оценивается:

- влияние ивентов на модели, текстуры, пост-процесс/цвет/свет, UI, звук, VFX, использование новых уникальных ассетов;
- общее качество визуальной коммуникации начала и завершения, качества ассетов, общей реализации с художественной точки зрения.

**Общий формат сдачи:**

- видео (строго до 2 мин), детально демонстрирующее цикл смены ивентов;
- билд.

**Правила оформления ответа на задания:**

- pdf-файл с таблицей;
- два видео.

Кроме того, проверяется наличие в билде.

## ***Решение***

Чтобы выполнить задачи, необходимо адаптировать визуальную часть игры под строгие требования гайдлайна. Это означает, что следует провести ревизию всех визуальных компонентов (интерфейса, моделей, эффектов и т. д.) и привести их к единому стилю и уровню качества, не создавая нового контента. При этом важно ориентироваться не только на эстетическую сторону, но и на удобство восприятия — сделать игру понятной визуально даже без текста. Если какие-то элементы визуально перегружены, нечитабельны или выбиваются из общего языка игры — они удаляются или перерабатываются. После визуального выравнивания всей игры нужно записать короткое видео, которое покажет:

- как выглядит интерфейс;
- как ведет себя игра;
- как игрок взаимодействует с визуальной частью.

В этом видео важно не говорить, а показать, что визуальная система чиста, читаема и соответствует заявленному стилю.

### ***Соответствие гайдлайну (обязательное задание)***

Ссылка на видео:<https://disk.yandex.ru/d/jCQqW5pyIQPKvw>.

### ***Формирование процедурной генерации в игре (задание на выбор)***

Ссылка на документ: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133800-Gv6E5PPCrt.pdf>.

### ***Глобальные ивенты (задание на выбор)***

Ссылка на видео:<https://disk.yandex.ru/d/jCQqW5pyIQPKvw>.

### ***Задача 4.3.5.3. Блок Программирование (100 баллов)***

#### ***Пользовательский контент (обязательное задание)***

Создайте функционал User Generated Content (далее UGC) для существующей игры.

Цель задания — реализовать возможность, позволяющую пользователям создавать и добавлять в игру собственные материалы, такие как: уровни, персонажи, предметы и другие элементы, способные расширить игровой процесс.

При выполнении задания:

- добавьте возможность импорта и экспорта дополнений, чтобы игроки могли ими делиться;
- продемонстрируйте свои примеры дополнений (не менее трех), чтобы показать возможности вашего UGC; учтите: в случае, если не реализована возможность импорта дополнений в игру, проверяющий не сможет проверить качество и разнообразие приложенных примеров UGC;
- предоставьте руководству по созданию UGC, которое позволит пользователям создавать свои моды; проверяющий будет использовать данное руководство для проверки задания (оно должно быть доступно и изнутри самой игры и включать в себя все необходимые инструкции об импорте, экспорте, создании, изменениях дополнений, загрузке своих файлов и т. д.); включите в руководство ошибки, которые могут возникнуть при работе пользователя с системой UGC, а также описание того, как их исправить (если возможно).

У игроков должна быть возможность:

- изменять внутриигровые параметры для создания дополнений (например, внутриигровые диалоги с NPC, скорость появления квестов и т. д.);
- загружать свои файлы для создания дополнений (например, png-картинки для текстур, свои модельки для NPC и т. д.)

Запишите видео с реализацией всех сделанных вами критериев (рекомендовано озвучить или добавить небольшие поясняющие субтитры).

Если проверяющий не сможет найти реализацию критерия внутри игры (неинтуитивно сделан функционал или другие случаи, в которых невозможно будет проверить, правильно ли выполнен критерий), то в ситуации, когда реализация содержится в видео, будет засчитана половина баллов за данный критерий с округлением в меньшую сторону ( $\frac{5}{2} = 2,5 \approx 2$ ).

Баллы за руководство и примеры дополнений не могут быть засчитаны через видео.

#### **Общий формат сдачи:**

- User Guide по созданию UGC (pdf-файл);
- ссылка на zip-файл (который должен находиться в предоставленном вам гугл-диске) с примером как минимум трех разных дополнений;
- видео (строго до 5 мин), демонстрирующее реализацию выполненных критериев.

### ***Процедурная генерация (задание на выбор)***

Разработайте систему процедурной генерации уровней для игры, используя сиды (seed). Система должна создавать разнообразные, но сбалансированные уровни, включая мир, задания, квесты и т. д. Сложность должна адаптироваться к уровню игрока, динамически изменяя параметры генерации, при этом иметь крайние значения (например, чтобы враги полностью не исчезли):

- Ручной ввод сида: у пользователя должна быть возможность вручную ввести сид, и система должна сгенерировать мир на его основе. При вводе одинаковых сидов на разных устройствах результат одинаковый.
- Генерация мира, заданий, квестов и т. д.: разработайте алгоритм генерации игрового мира и других элементов игры на основе сида. Мир должен быть разнообразным (ландшафты, расположение объектов) и сохранять определенную степень баланса (например, равномерное распределение ресурсов).
- Адаптивная сложность: реализуйте механизм адаптации сложности генерации к прогрессу игрока, например, увеличьте плотность врагов или сложность заданий после успешного прохождения квестов или прохождения игры и, наоборот, уменьшите количество врагов, если игрок долго не может пройти определенный уровень.

Предоставьте блок-схему этого алгоритма, создайте ее на сайте draw.io (<http://draw.io/>).

Если проверяющий не сможет найти реализацию критерия внутри игры (неинтуитивно сделан функционал или другие случаи, в которых невозможно будет проверить, правильно ли выполнен критерий), то в ситуации, когда реализация содержится в видео, будет засчитана половина баллов за данный критерий с округлением в меньшую сторону ( $\frac{5}{2} = 2,5 \approx 2$ ). Баллы за руководство и примеры дополнений не могут быть засчитаны через видео.

### **Общий формат сдачи:**

- блок-схема системы генерации (созданная на сайте draw.io с форматом файла `.drawio`);
- видео (строго до 5 мин), демонстрирующее реализацию выполненных вами критериев.

## **Глобальные иVENTы (задание на выбор)**

Разработайте систему глобальных ивентов, используя взаимодействие с сервером со второго этапа. Система должна управлять ивентами различной длительности (ежедневные, еженедельные и т. д.), влиять на разные части игры, а также уведомлять игроков о предстоящих ивентах.

Для взаимодействия с сервером обязательно использовать данную инструкцию:  
<https://disk.yandex.ru/i/MD88qfhWUusJSA>.

При разработке системы учитывайте следующие моменты:

- Серверная система ивентов: создайте ивенты на сервере и реализуйте механизм их получения не чаще, чем 1 раз в 2 мин. Обеспечьте, чтобы ивенты не пересекались по времени, если они влияют на одни части игры. Постарайтесь назвать ивенты так, чтобы их названия на сервере были соотносимы с ивентами внутри игры.
- Одновременность: одни и те же ивенты должны запускаться на разных устройствах одновременно (даже на разных часовых поясах), допуская небольшую разницу во времени.
- Клиентская система уведомлений: система должна отображать уведомления игроку о предстоящих событиях (например, за день для еженедельных и за 15 мин для внутридневных).
- Влияние на геймплей и UI: каждое событие должно оказывать заметное влияние на игровой процесс и интерфейс пользователя. Влияние должно быть визуально понятно и соответствовать типу события.
- Защита от ошибок: реализуйте обработку ошибок при запросе к серверу, например, при отсутствии интернета. Ивент в любом случае должен произойти с использованием старых данных.
- Динамическое изменение: ивенты должны подстраиваться под изменение данных на сервере. Проверяющий будет изменять время ивентов, чтобы они произошли во время проверки. Желательно заранее подстроить время ивентов, чтобы оно выпало на день и время проверки (с 27 марта 17:00 до 28 марта 04:00). В случае, если ивенты не будут изменяться динамически или не будут подстроены под время проверки, может возникнуть ситуация, когда проверяющий не увидит ивенты или уведомления, поэтому баллы придется выставлять по видео.

Предоставьте блок-схему этого алгоритма, создайте ее на сайте draw.io (<http://draw.io/>). Запишите видео с реализацией всех сделанных критериев (рекомендовано озвучить или добавить небольшие поясняющие субтитры).

Если проверяющий не сможет найти реализацию критерия внутри игры (неинтуитивно сделан функционал или другие случаи, в которых невозможно будет проверить, правильно ли выполнен критерий), то в ситуации, когда реализация содер-

жится в видео, будет засчитана половина баллов за данный критерий с округлением в меньшую сторону ( $\frac{5}{2} = 2,5 \approx 2$ ). Баллы за руководство и примеры дополнений не могут быть засчитаны через видео.

### **Общий формат сдачи:**

- блок-схема системы генерации (созданная на сайте draw.io с форматом файла **.drawio**);
- видео (строго до 5 мин), демонстрирующее реализацию выполненных вами критериев.

### **Правила оформления ответа на задания:**

Весь ответ:

- User Guide по созданию UGC (pdf-документ);
- ссылка на zip-файл (должен находиться в предоставленном вам гугл диске) с примером минимум трех разных дополнений;
- два видео (строго до 5 мин), демонстрирующие реализацию выполненных критериев.
- блок-схема задания на выбор (созданная на сайте draw.io с форматом файла **.drawio**).

Проверяющий при контроле заданий блока Программирование будет использовать билд, прикрепленный в блоке Продукт.

### ***Решение***

Чтобы выполнить задачи этого блока, необходимо сосредоточиться на реализации пользовательского контента (UGC), процедурной генерации или глобальных ивентов.

Если выбран UGC, то в первую очередь создается система, позволяющая игрокам самостоятельно добавлять в игру контент — уровни, предметы, персонажей. Это требует настройки импорта/экспорта, интерфейса для настройки параметров, а также — системы обработки ошибок. Нужно подготовить три примера UGC, которые можно загрузить в игру, а кроме того — подробный гайд с описанием того, как этим пользоваться, с пояснением возможных багов и их решений.

Если выбран процедурный блок, то проектируется система генерации на базе сида с балансировкой и адаптацией сложности. Сид должен гарантировать повторяемость, а сама генерация — разнообразие при сохранении логики.

В дополнение к алгоритму делается блок-схема (например, в draw.io) и видео с демонстрацией работы генерации. Аналогично, если используется блок ивентов, создается серверная и клиентская логика, реализуются уведомления и адаптивные реакции в игре, записывается видео и предоставляется схема логики работы событий. В каждом случае основной результат — не просто рабочая фича, а ее наглядная и понятная демонстрация.

## **Пользовательский контент (обязательное задание)**

Юзер-гайд по созданию UGC (pdf-файл): <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133428-49jAe8GUrK.pdf>.

Пример 3 разных дополнений в zip-файле: <https://disk.yandex.ru/d/QgndHaG0Ohofeg/mods.zip>.

Ссылка на видео (до 5 мин): [https://disk.yandex.ru/d/QgndHaG0Ohofeg/Prog\\_Osnovn.mov](https://disk.yandex.ru/d/QgndHaG0Ohofeg/Prog_Osnovn.mov).

## **Процедурная генерация (задание на выбор)**

Блок-схема (созданная на сайте draw.io с форматом файла .drawio): <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327133347-0oMGU6yq0G.drawio>.

Ссылка на видео: [https://disk.yandex.ru/d/QgndHaG0Ohofeg/Prog\\_Generation.mov](https://disk.yandex.ru/d/QgndHaG0Ohofeg/Prog_Generation.mov).

## **Глобальные ивенты (задание на выбор)**

Блок-схема (созданная на сайте draw.io с форматом файла .drawio): <https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327132258-7bnMdajOJj.drawio>.

Ссылка на видео: <https://disk.yandex.ru/i/3FL21sTXfuH12g>.

## **Задача 4.3.5.4. Блок Продукт (150 баллов)**

### **Трейлер (обязательное задание)**

Создание трейлера для игры — это творческий процесс, который требует внимания к деталям, навыков видеомонтажа и глубокого понимания целевой аудитории. Главная задача — создать трейлер, который ярко и убедительно представит игру, выделив ее ключевые достоинства.

При создании трейлера не забудьте:

- о том, что трейлер должен соответствовать целевой аудитории, отображать жанр игры;
- добавить название команды и игры в начале трейлера (креативность появления приветствуется);
- использовать качественные, достоверные и разнообразные материалы;
- подобрать подходящую музыку и звуки, которые передают настроение игры.

Важно: продолжительность трейлера должна составлять от 40 с до 1 мин 30 с.

**Общий формат сдачи:** видео (строго до 90 с).

## ***Презентация и выступление (обязательное задание)***

Пришло время представить игру инвесторам! Для этого нужно объединить все наработки в одну презентацию и подготовить питч. Используйте предоставленный шаблон для создания презентации, тщательно подготовьтесь к выступлению и будьте готовы ответить на вопросы экспертов.

Важно: общее время выступления, включая ответы на вопросы, не должно превышать 10 мин. Уделите внимание ключевым аспектам проекта, чтобы заинтересовать аудиторию и четко донести свои идеи. Удачи!

Перед началом презентации покажите трейлер игры!

Презентация должна включать в себя:

- Название команды и игры.
- Паспорт игры.
- Анализ конкурентов.
- Описание геймплея.
- Анализ целевой аудитории.
- Описание выбранного стиля.
- Уникальность игры.
- Масштабируемость игры.
- Участники команды.

Ссылка на шаблон: <https://disk.yandex.ru/d/eXjzHrYDmlMCpA>.

При желании можно использовать свое оформление!

**Общий формат сдачи:** презентация.

## ***Страницка магазина (обязательное задание)***

Задача — оформить страницу игры в магазине. Она должна быть информативной, привлекательной и соответствовать всем требованиям.

На страницу необходимо добавить:

- Название игры.
- Логотип игры.
- Баннер игры.
- Наличие краткого описания (до 40 слов).
- Наличие полного описания (от 100 до 250 слов).
- Скриншоты из игры, которые отражают геймплей.
- Имя и фамилия разработчиков проекта, с указанными ролями и фотографиями.
- Дорожная карта развития проекта (визуализация реалистичной стратегии развития игры после финала НТО).

Важно: при заполнении следите за единообразием страницы и отсутствием ошибок в тексте.

Для упрощения разработан шаблон, который нужно использовать: <https://www.figma.com/file/b0XIpfvW4hDjWKNGb58yOV/Mocup-RuStore?type=design&node-id=0-1&mode=design&t=wYjR9C2AKmFpU8vX-0>.

### **Общий формат сдачи:**

- ссылка на Figma;
- pdf-файл Figma;
- zip-файл, лого в png, скрины из игры, краткое и полное описание игры в блокноте с Ф.И.О. разработчиков, дорожная карта.

## ***Билд игры (обязательное задание)***

Задача — подготовить финальную версию билда игры и отправить ее для проверки. Для этого необходимо загрузить файл игры (файлы для ОС Windows) в вашу папку на гугл-диске.

Убедитесь, что финальный билд полностью готов к проверке и соответствует всем требованиям. После завершения тестирования и исправления ошибок загрузите файлы в указанную папку.

При проверке билда делается акцент на том, насколько:

- игра захватывает игрока, создает ощущение присутствия в игровом мире;
- качественно, разнообразно и сбалансировано реализован основной игровой процесс (Соге-геймплей);
- продуман и увлекательен мета-геймплей (прогресс персонажа, прокачка, коллекционирование и т. д.);
- хорошо проработана игровая среда и мир;
- в игре присутствуют элементы, побуждающие к повторному прохождению;
- стабильно работает игра;
- качественно продуман и удобен игровой интерфейс;
- хорошо реализована система управления;
- качественно выполнена графика и визуальные эффекты;
- проработаны звуковые и музыкальные эффекты;
- хорошо игра обучает игрока.
- реализована система наград и мотивации.

**Общий формат сдачи:** билд (файл для ОС Windows с игрой на гугл-диск).

**Правила оформление ответа на задания:** весь ответ — одно видео (до 90 с), одна презентация, ссылка на заполненную Figma, pdf с Figma, zip-файл, билд (файл для ОС Windows с игрой на гугл-диске).

## ***Решение***

Для решения задач блока нужно подготовить игру как продукт для внешнего пользователя — инвестора, платформы или магазина. Все начинается с трейлера: короткое, но цепляющее видео, которое демонстрирует атмосферу игры и ее геймплей. Затем подготавливается презентация. Она создается по шаблону, но ее зада-

ча — вызвать интерес, а не просто информировать. Здесь описывается проект как бизнес-единица, поэтому важно дать ответы на вопросы:

- В чем идея проекта?
- Что представляет собой ЦА?
- Как проект масштабируется?
- Кто входит в команду?
- Кто конкуренты?

Следом оформляется страница магазина (по шаблону RuStore) с логотипом, баннером, коротким и полным описанием, скриншотами, фото и ролями разработчиков, а также — дорожной картой проекта на полгода-год вперед. Здесь важна не только эстетика, но и полнота: страница должна выглядеть так, как будто игру уже можно продавать.

Все это сопровождается билдом: стабильной, законченной версией игры, которая демонстрирует все ключевые механики, интерфейс, обучение, визуал и звук. Содержание билда должно соответствовать всему, что заявлено в материалах. Все они упаковываются и подаются как единый продукт, готовый к продвижению.

### ***Трейлер (обязательное задание)***

<https://disk.yandex.ru/d/6T811jADH-ewrQ>.

### ***Страница магазина (обязательное задание)***

<https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327153218-TMPLXGSFQ1.txt>.  
<https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327153230-YxLDC1WR0P.pdf>.  
<https://disk.yandex.ru/d/QgndHaG0Ohofeg/20250327153518-Wz8sXKA8d7.zip>.

### ***Билд игры (обязательное задание)***

[https://disk.yandex.ru/d/QgndHaG0Ohofeg/Arktos\\_Final.zip](https://disk.yandex.ru/d/QgndHaG0Ohofeg/Arktos_Final.zip).

### ***Задача 4.3.5.5. Блок Берлога (50 баллов)***

#### ***Игровой туториал (обязательное задание)***

В видеоиграх, особенно в симуляторах, обучение (*tutorial*) играет ключевую роль. Оно позволяет игрокам быстро освоиться с управлением, разобраться в основных механиках и погрузиться в игровой процесс.

Хорошее обучение — это не просто сухой набор инструкций, а увлекательное введение в мир игры, которое мотивирует игрока продолжать. Обучение помогает игроку быстрее и внимательнее пройти уровни, ведет к достижению целей игры,

заложенных разработчиками, а в полезных играх среди таких целей — еще и получение новых знаний.

В успешных играх тьюториалы позволяют формировать учебную субъектность игрока, когда он самостоятельно ставит себе задачи для изучения в пространстве игры, и эти задачи помогают ему в прохождении.

### **Постановка задачи**

Спроектируйте и реализуйте игровой тьюториал для игры.

Обучение должно быть:

- Интерактивным — игрок должен учиться, выполняя действия, а не просто читая текст.
- Постепенным — новые механики и элементы управления должны нужно вводить последовательно, не перегружая игрока.
- Наглядным — визуальные подсказки, подсветку, анимации, указатели и т. д. помогают сделать его максимально понятным.
- Уместным — обучение должно органично вписываться в игровой процесс и соответствовать сюжету игры.
- Дружелюбным — обучение должно быть позитивным, поощрять игрока за успехи и давать возможность повторить сложные моменты.
- Полезным — игрок получает полезные для игрового процесса навыки и знания.

### **Правила оформления ответа на задания**

Предоставьте видеоролик длительностью не более 5 мин, иллюстрирующий систему игрового тьюториала.

Ролик должен включать демонстрацию:

- обучения (то есть того, как игрок взаимодействует с обучением, как вводятся новые элементы управления и механики, как выглядят визуальные подсказки и т. д.);
- геймплея после прохождения обучения.

Важно: можно добавить закадровый голос или текстовые комментарии, поясняющие происходящее в игре.

### **Интро (обязательное задание)**

Интро (вступительный ролик) в игре — элемент повествования и погружения в игровой мир, способ рассказать о сеттинге. Это первое, что видит игрок, запуская игру, и от качества интро зависит, заинтересуется ли он, захочет ли узнать историю и погрузиться в игровой процесс.

Интро выполняет несколько ключевых функций:

- знакомит с персонажами;
- обрисовывает сюжетную завязку;
- обозначает основной конфликт;
- создает нужную атмосферу.

В хорошем интроверте сжато и емко представлена экспозиция, которая готовит игрока к игре. Интроверт — часть игры как форма медиа, которая поможет в дальнейшем ее продвижению среди аудитории.

### ***Постановка задачи***

Создайте и интегрируйте интроверт в игру.

Интроверт должно:

- познакомить с сеттингом — показать, где и в какое время происходят действия игры;
- представить ключевых персонажей — продемонстрировать хотя бы двух персонажей и их традиций, которые будут играть важную роль в сюжете; здесь можно коротко раскрыть их характеры, мотивацию, взаимоотношения;
- обозначить конфликт — показать проблему, с которой столкнутся персонажи, а игроку предстоит помочь в ее разрешении;
- заинтересовать и вовлечь — интроверт должно быть динамичным, интригующим, вызывать желание узнать, что будет дальше;
- соответствовать жанру и стилю игры — нужно, чтобы интроверт гармонировало с общей стилистикой и атмосферой игры-симулятора.

### ***Правила оформления ответа на задания***

Предоставьте видеоролик длительностью не более 60 с, демонстрирующий интроверт к вашей игре. Интроверт должен быть интегрирован в игру.

Технические требования к видео:

- формат — MP4, AVI, MOV или другой распространенный видеоформат;
- разрешение — не менее  $1280 \times 720$  (720p);
- качество — видео должно быть четким, а звук разборчивым.

### ***Решение***

Для выполнения заданий этого блока необходимо сфокусироваться на обучении игрока и нарративной подаче.

Туториал — это не инструкция, а часть игрового опыта. Он должен быть встроен в геймплей, быть интерактивным, последовательным и визуально объясняющим, что именно нужно делать. Туториал должен научить игрока основам без перегрузки, поощрять за успехи и позволять повторить ошибки.

Не менее важно сделать интроверт — вступительную сцену или ролик, который погружает в сеттинг, показывает ключевых персонажей, их мотивацию и основной конфликт. Интроверт может быть как катсценой, так и набором игровых событий, но оно должно быть увлекательным, динамичным и понятным.

Видеодемонстрации туториала и интроверта должны быть четкими, без лишних слов, акцентирующими внимание на важном.

В результате блок Берлога показывает, как игра встречает нового игрока — это его первый опыт, и он должен быть захватывающим.

## ***Игровой туториал (обязательное задание)***

<https://disk.yandex.ru/d/QgndHaG0Ohofeg/tutorial.mp4>.

## ***Интро (обязательное задание)***

<https://disk.yandex.ru/i/Vnz1QwzvnRzztw>.

### **4.3.6. Система оценивания**

#### ***Процедура оценки результатов***

Блоки Арт, Программирование, Геймдизайн оцениваются группой экспертов заочно по санным решениям. Максимальная оценка по каждому из блоков — **100 баллов**.

Блок Берлога также оценивается группой экспертов заочно по санным решениям. Максимальная оценка по блоку — **50 баллов**.

Часть блока Продукт оценивается группой экспертов заочно, а именно Трейлер, Страница. Максимальная оценка — **50 баллов**.

Если решение сдано после дедлайна, применяется дисконтирование в 50%.

В заключительный день проводится очная оценка проектов в присутствии команд и экспертов игровой индустрии.

**Первый этап оценки** проводится в формате шоукейсов<sup>2</sup> среди всех команд.

На каждую командудается 5 мин для демонстрации работоспособности прототипа и ответа на вопросы экспертов. Несколько групп экспертов могут параллельно производить оценку команд (т. е. в зависимости от желания экспертов они могут работать одной группой, по одному или объединяться в меньшие группы, но каждый эксперт должен оценить все 17 команд).

Максимальная оценка по итогу шоукейсов — **60 баллов**. Итоговый балл по данному блоку вычисляется как среднее арифметическое баллов всех экспертов, участвующих в оценке. При возникновении сомнений со стороны экспертов участники должны суметь обосновать самостоятельность выполнения проекта.

Таким образом, после проведения первого этапа финального испытания шоукейсов и оценки блоков Программирование, Арт, Геймдизайн и Берлога максимальная оценка, которую может получить команда — **460 баллов**.

Рейтинговая таблица формируется согласно оценке этих блоков. После проведения первого этапа финального испытания она публикуется в чате всех команд с разбивкой по критериям. **Первые семь команд, набравшие наибольшее количество баллов, допускаются до защиты презентаций перед экспертами.**

<sup>2</sup>Шоукейсы — это возможность рассказать о своей игре и презентовать продукт; в нашем случае — это демонстрация играбельности прототипа.

Второй этап представляет собой презентацию семи лучших команд экспертному жюри. Каждой команде отводится 10 мин на демонстрацию своего проекта согласно шаблону и ответы на вопросы экспертов.

Итоговым баллом по данному блоку является среднее арифметическое баллов всех экспертов, участвующих в оценке. При возникновении сомнений со стороны экспертов участники должны суметь обосновать самостоятельность выполнения проекта.

Максимальная оценка, которую команды могут получить за презентации проектов — **40 баллов**.

**Баллы, полученные за презентацию, учитываются только в командном зачете, но не влияют на итоговый индивидуальный результат участника.**

Таким образом, максимальная командная оценка за решение задач инженерного тура составляет **500 баллов**. Побеждает команда, набравшая наибольшее количество баллов.

В случае равенства баллов и невозможности определить команду-победителя дополнительно могут оцениваться качество ведения репозиториев, а именно, организация структуры проектов.

## **Блок Геймдизайн (100 баллов)**

Таблица 4.3.3. Критерии оценивания

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реализацией</b>	<b>экспертной оценки</b>
<b>Задание 1. Соответствие методологии</b>	<b>30</b>	<b>20</b>
Выбор видов вовлеченности и описание существующих механик, работающих на его получение: 0 — выбраны не две формы вовлеченности или формы вовлеченности не соответствуют представленным в ресурсе; 3 — описано 0–1 механика, направленная на достижение этого вида вовлеченности, описание исчерпывающее и не содержит логических ошибок; 6 — описано больше двух механик, направленных на достижение этого вида вовлеченности, но в описании достижении вовлеченности допущены логические ошибки; 10 — описано больше двух механик, направленных на достижение этого вида вовлеченности, описание исчерпывающее и не содержит логических ошибок.	10	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Определение столпов игры и приведение существующих механик, дополняющих их:  0 — описано не три–четыре ключевые геймплейные механики (КГМ) или хотя бы одна механика не является ключевой, исходя из геймплея игры или хотя бы одна механика не соотносится с выбранными ранее видами вовлеченности; 3 — описано три–четыре КГМ, описано не более одной дополнительной механики к каждой КГМ или более одной дополнительной механики, но описание связи содержит логические ошибки, связь дополнительных механик неочевидно описана; 6 — описано три–четыре КГМ, описано более одной дополнительной механики к каждой КГМ, описание связи содержит минорные логические ошибки; 10 — описано три–четыре КГМ, описано более одной дополнительной механики к каждой КГМ, описание связи не содержит логических ошибок.	10	0
Фичекат.  Обоснованность удаления/улучшения механик: 0 — определено менее двух механик для вырезания; 5 — определено более двух механик, необъективное обоснование; вырезаны действительно филлерные механики; имеются отхождения от целевых видов вовлеченности; 10 — определено более двух механик, обоснование содержит минорные логические проблемы; вырезаны действительно филлерные механики, улучшение допускает некритичные отхождения от целевых видов вовлеченности; 15 — определено более двух механик, обоснование не содержит логических проблем; вырезаны действительно филлерные механики, улучшение допускает минорные отхождения от целевых видов вовлеченности.	5	10
Видео с изменениями геймплея. Целостность игрового процесса.  Темп игры: 0 — менее двух механик вырезано или видео дольше 3 мин; 5 — определено более двух механик, на видео видно их изменение/удаление, но заметно критичное проседание в темпе игры; 10 — определено более двух механик, на видео видно их изменение/удаление, но заметно проседание в темпе игры, или замена в сегменте не соответствует видам вовлеченности и ключевым механикам; 15 — определено более двух механик, на видео видно их изменение/удаление, заметно лишь минорное проседание в темпе игры и замена в сегменте соответствует видам вовлеченности и КГМ.	5	10
<b>Итого</b>	<b>50</b>	
<b>Задание 2.</b>	<b>30</b>	<b>20</b>

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<p><b>Обоснование необходимости процедурной генерации в игре</b>  Необходимо подробно раскрыть все три проблемы. При этом прописаны три аспекта влияния, которые улучшают текущую игровую концепцию и не противоречат ей, и два способа интеграции в лор не противоречат и дополняют лор:</p> <p>0 — выполняются три пункта:</p> <ul style="list-style-type: none"> <li>● один или более аспектов не влияют на геймплей;</li> <li>● один или более представленных способов не подходят лору и атмосфере проекта; или отсутствует один из ключевых пунктов (например, описано только две проблемы или один способ интеграции в лор).</li> </ul> <p>2 — выполняются два пункта:</p> <ul style="list-style-type: none"> <li>● процедурная генерация не решает хотя бы одну из прописанных проблем;</li> <li>● процедурная генерация не решает хотя бы одну из прописанных проблем;</li> <li>● один или более аспектов не влияют на геймплей;</li> <li>● один или более представленных способов не подходят лору и атмосфере проекта.</li> </ul> <p>4 — выполняется один пункт:</p> <p>Процедурная генерация не решает хотя бы одну из прописанных проблем.</p> <ul style="list-style-type: none"> <li>● один или более аспектов не влияют на геймплей;</li> <li>● один или более представленных способов не подходят лору и атмосфере проекта.</li> </ul> <p>8 — подробно раскрыты все три проблемы, три аспекта влияния улучшают текущую игровую концепцию и не противоречат ей, два способа интеграции в лор не противоречат и дополняют лор.</p>	8	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<p><b>Элементы процедурной генерации</b>  Необходимо указать не менее трех элементов генерации. Для каждого элемента описаны механики, способные генерировать не менее 10 уникальных единиц контента (всего 30).  Приведены три примера уникальных единиц контента для каждого элемента (всего девять). Все получившиеся и потенциальные единицы контента обладают уникальными свойствами и игровыми механиками.</p> <p>0 — больше пяти получившихся и потенциальных единиц контента соответствуют хотя бы одному пункту ниже:</p> <ul style="list-style-type: none"> <li>● не соответствуют основным игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● не вписываются в лор или сеттинг,</li> </ul> <p>или отсутствует один из ключевых пунктов (например, описано только два элемента генерации или генерируется только 10 уникальных единиц контента).</p> <p>2 — до 10 получившихся или потенциальных единиц контента в точности повторяют свойства и механики друг друга, то есть не являются уникальными.</p> <p>4 — до пяти получившихся и потенциальных единиц контента соответствуют хотя бы одному пункту ниже:</p> <ul style="list-style-type: none"> <li>● не соответствуют основным игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● не вписываются в лор или сеттинг.</li> </ul> <p>8 — указаны не менее трех элементов генерации. Описаны механики генерации для каждого элемента, которые могут генерировать не менее десяти уникальных единиц контента (всего 30).  Приведены три примера уникальных единиц контента для каждого элемента (всего девять).</p> <p>Все получившиеся и потенциальные единицы контента обладают уникальными свойствами и игровыми механиками.</p>	8	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<p><b>Контроль случайности</b>            Необходимо описать три механики балансировки случайности, два правила генерации и два метода контроля сложности. Все механики должны логично вписываться в концепцию проекта, не противоречить основным игровым механикам и улучшать игровой баланс.</p> <p>0 — от пяти механик, правил или методов управления случайностью подходят под один из пунктов:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● неэффективны в контроле сложности; или отсутствует один из ключевых пунктов (например, описаны только две механики или один метод контроля сложности).</li> </ul> <p>2 — от двух до четырех механик, правил или методов управления случайностью подходят под один из пунктов:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● неэффективны в контроле сложности.</li> </ul> <p>4 — до двух механик, правил или методов управления случайностью подходят под один из пунктов:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● неэффективны в контроле сложности.</li> </ul> <p>8 — описаны три механики балансировки случайности, два правила генерации, два метода контроля сложности. Все механики логично вписываются в концепцию проекта, не противоречат основным игровым механикам и улучшают игровой баланс.</p>	4	4

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<p><b>Взаимодействие игрока с процедурной генерацией</b>        Необходимо описать три способа влияния, два тактических преимущества и две механики антифрустрации. Все способы должны быть интегрированы в игровой процесс, не нарушать баланс и улучшать вовлеченность игрока.</p> <p>0 — от четырех способов, механик или преимуществ оказывается повторяющимися или не имеющими смысла в текущей механике игры, или от четырех способов, механик или преимуществ соответствуют хотя бы одному пункту ниже:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● не имеют реального влияния на игру;</li> </ul> <p>или отсутствует один из ключевых пунктов (например, описаны только два способа влияния или один механика анти-фрустрации).</p> <p>2 — до трех способов, механик или преимуществ оказывается повторяющимися или не имеющими смысла в текущей механике игры, или до трех способов, механик или преимуществ соответствуют хотя бы одному пункту ниже:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● не имеют реального влияния на игру.</li> </ul> <p>4 — до двух способов, механик или преимуществ оказывается повторяющимися или не имеющими смысла в текущей механике игры; или до двух способов, механик или преимуществ соответствуют хотя бы одному пункту ниже:</p> <ul style="list-style-type: none"> <li>● не соответствуют игровым механикам;</li> <li>● противоречат игровой концепции;</li> <li>● не имеют реального влияния на игру.</li> </ul> <p>8 — описаны три способа влияния, два тактических преимущества и две механики анти-фрустрации. Все способы интегрированы в игровой процесс, не ломают баланс и улучшают вовлеченность игрока.</p>	4	4

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<b>Описание первой части игры и видеодемонстрация процедурной генерации</b> Полностью описаны три игровых опыта игрока. Все элементы дизайна из пункта «Элементы процедурной генерации» интегрированы в игровой процесс, не ломают баланс и улучшают вовлеченность игрока. Показаны не менее трех различных элементов сгенерированного контента (по пять единиц на каждый элемент). Явно продемонстрирована работа генерации. 0 — выполняется один из пунктов: <ul style="list-style-type: none"><li>● описано менее трех игровых опыта игрока;</li><li>● все описанные элементы дизайна могут ломать баланс, а также негативно сказываться на игровой опыт игрока;</li><li>● показан только один элемент сгенерированного контента с менее пяти единиц на каждый элемент;</li><li>● хотя бы один продемонстрированный элемент процедурной генерации отличается от описанного в пункте «Элементы процедурной генерации»;</li><li>● работа случайной генерации не продемонстрирована.</li></ul> 4 — выполняется один из пунктов: <ul style="list-style-type: none"><li>● только один описанный элемент дизайна интегрирован в игровой процесс;</li><li>● показан только один элемент сгенерированного контента по пять единиц на каждый элемент.</li></ul> 8 — Выполняется один из пунктов: <ul style="list-style-type: none"><li>● только два описанных элемента дизайна интегрированы в игровой процесс;</li><li>● показаны не менее двух различных элементов сгенерированного контента по пять единиц на каждый элемент;</li><li>● работа случайной генерации продемонстрирована лишь частично.</li></ul> 12 — полностью описаны три игровых опыта игрока. Все описанные элементы дизайна в пункта «Элементы процедурной генерации» интегрированы в игровой процесс: <ul style="list-style-type: none"><li>● до двух из описанных элементов дизайна, интегрированных в игровой процесс, могут ломать баланс, а также негативно сказываться на игровой опыт игрока;</li><li>● показаны не менее трех различных элементов сгенерированного контента по пять единиц на каждый элемент;</li><li>● явно продемонстрирована работа случайной генерации.</li></ul> 18 — полностью описаны три игровых опыта игрока. Все описанные элементы дизайна в пункте «Элементы процедурной генерации» интегрированы в игровой процесс, не ломают баланс и улучшают вовлеченность игрока. Показано не менее трех различных элементов сгенерированного контента по пять единиц на каждый элемент. Явно продемонстрирована работа случайной генерации.	6	12
<b>Итого</b>	<b>50</b>	
<b>Задание 3. Глобальные ивенты</b>	<b>30</b>	<b>20</b>

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Подходит под ЦА. Обоснование приведено для всех вариантов ивентов в игре. Аргументы отвечают на вопросы: «Почему моей игре подходит этот вид временных событий?», «Как это временное событие поможет увеличить retention в моей игре?» 0 — описание имеет неоднозначные формулировки или не приведено достаточной аргументации; 1 — есть минорные проблемы в описании некоторых (не всех) типов, но в целом аргументация логична и правильна; 2 — качественная аргументация приведена для всех вариантов ивентов в игре.	1	1
Декомпозиция ивентов у игр. Декомпозиция содержит все важные в контексте задания элементы. Для каждого из примеров при декомпозиции упущены: 0 — важные и критичные для опыта игрока элементы; 1 — важный элемент системы; 2 — элемент системы, влияющий на эмоции игрока непосредственно. 3 — При декомпозиции не упущено важных в контексте задания элементов.	2	4
Обоснование приемлемости в игре.	1	0
Обоснование выбора. Поиск узких мест для обоснованного отказа: 0 — в предыдущем пункте «Обоснование приемлемости в игре» дано обоснование не на полный балл или обоснование имеет в себе значительные ошибки; 1 — обоснование имеет минорные ошибки; 2 — обоснование исчерпывающее и приведены объективные доводы для отказа от выбранной в этом пункте механики.	2	0
Описание системы событий должно включать разнообразные виды контента и способы их получения. Чем больше уникальных, проработанных механик и вариативности в подаче материалов, тем выше оценка. Для каждого из примеров: 0 — более четырех видов контента, более трех видов получения контента; 2 — четыре–пять видов контента, три–четыре вида получения контента; 3 — четыре–пять видов контента, три–четыре вида получения контента, механики оригинальны; 4 — больше пяти видов контента, больше четырех видов получения контента, механики оригинальны.	4	4
Пример контента должен быть приведен и иметь потенциал для вовлечения. Для каждого из примеров: 0 — пример контента не приведен; 1 — контент имеет потенциал для завлечения; 2 — описанный контент интересен.	2	2

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<p>Влияние на геймплей должно быть раскрыто через несколько примеров.</p> <p>Для каждого из примеров:</p> <ul style="list-style-type: none"> <li>0 — приведено меньше двух описаний;</li> <li>1 — приведено больше одного описания;</li> <li>2 — приведено больше двух описаний и хотя бы два из них не состоят не только из добавлений к статам/визуалу.</li> </ul>	4	0
<p>Блок-схема. Полнота отображения пути игрока по событию.</p> <p>Для каждого из примеров:</p> <ul style="list-style-type: none"> <li>0 — блок-схема плохо читается или несодержательна;</li> <li>1 — схема не ощущается окончательной, пропущены довольно важные пункты взаимодействия;</li> <li>2 — схема полно отражает путь игрока по событию, легко читается, все важные пункты занесены.</li> </ul>	4	0
<p>Механика должна быть четко считываемой на видео.</p> <ul style="list-style-type: none"> <li>0 — если за предыдущий пункт «Блок-схема» стоит 0 баллов или некоторые тайм-коды не соотносятся с содержанием видео;</li> <li>2 — некоторые важные моменты из п. 3.4 не показаны на видео;</li> <li>3 — неочевидная обратная связь от механик, тем самым проблемно считывается на видео;</li> <li>4 — показ механик позволяет однозначно определить этапы и результат взаимодействия с событием.</li> </ul>	4	0
<p>Создание мокапа интерфейса и UX-проработка. Интерфейс в видео должен быть показан.</p> <ul style="list-style-type: none"> <li>0 — отсутствие показа интерфейса или плохая навигация;</li> <li>2 — малое количество UI (одно окно или два–три окна с малым количеством взаимодействий) или наличие большого количества текста, там, где это не нужно;</li> <li>4 — отсутствие полноценных окон интерфейса(только окна с одним–двумя вариантами взаимодействия) или проблема с интуитивностью окон интерфейса;</li> <li>6 — интуитивный, приятно выглядящий (с учетом того, что это потенциально лишь макет) интерфейс.</li> </ul>	4	2
<p>Влияние механик событий на геймплей должно быть отображено в видео.</p> <ul style="list-style-type: none"> <li>0 — отсутствие явного показа влияния механик событий на игровой процесс или влияние состоит из только незначительных изменений, а не отдельных механик(в добавлении к статам, скрипты);</li> <li>2 — влияние показано, и оно значительно.</li> </ul>	2	0
<p>Структура должна быть удобной для восприятия и хорошо читаемой. Важно, чтобы информация подавалась ясно и наглядно.</p> <ul style="list-style-type: none"> <li>0 — выглядит непривлекательно, неудобно считывается;</li> <li>2 — удобно воспринимать информацию с дорожной карты.</li> </ul>	0	2

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Разнообразие событий представленных на дорожной карте: 0 — описано менее четырех событий; 1 — описано четыре–пять событий или различие между событиями минимально (только тематика, нарратив, числовые бонусы); 2–5 — описано шесть и более событий; механики в разных событиях повторяются с минимальными изменениями; 6 — каждое событие уникально, описаны новые механики или значительное преображение предыдущих.	1	5
<b>Итого</b>	<b>50</b>	

## **Блок Арт (100 баллов)**

Таблица 4.3.4. Критерии оценивания

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<b>Задание 1. Соответствие гайдлайну</b>		
Выполнены условия раздела «UI/UX»	5	5
Выполнены условия раздела «Шрифты и текста»	4	3
Выполнены условия раздела «Относительные размеры интерфейса на экране»	3	0
Выполнены условия раздела «Цвета»	3	2
Выполнены условия раздела «Аудио»	2	0
Выполнены условия раздела «Целостность визуального стиля»	0	8
Выполнены условия раздела «Текстуры/спрайты»	3	0
Выполнены условия раздела «Паттернализация/вариативность повторяющегося контента»	4	0
Выполнены условия раздела «Свет»	3	0
Выполнены условия раздела «Использование пост-процесса и паттернов»	3	0
Выполнены условия раздела «Анимация»	2	0
<b>Итого</b>	<b>50</b>	
<b>Задание 2. Процедурная генерация</b>		
Количество категорий элементов используемых в генерации: 3 категории — 1 балл; 10 категорий — 5 баллов; больше 20 категорий — 10 баллов.	10	0
Количество единиц контента (ассетов) используемых для генерации: 20 категорий — 1 балл.; 40 категорий — 5 баллов; больше 60 категорий — 10 баллов.	10	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Количество игровых элементов, созданных процедурно: 1 категория — 4 балла; 2 категории — 6 баллов; 3 категории — 8 баллов; 4 категории — 10 баллов.	10	0
Общее визуальное качество итогового генерируемого контента	0	20
<b>Итого</b>	<b>50</b>	
<b>Задание 3. Глобальные иVENTы</b>		
Глобальные иVENTы влияют на внешний вид игровых и декоративных элементов. Модели, текстуры, пост-процесс/цвет/свет, UI, звук, VFX — по пять баллов за каждый элемент	25	0
Качество реализации визуальной составляющей глобальных иVENTов (предыдущий пункт)	0	10
Четкая визуальная коммуникация начала и завершения глобальных иVENTов	0	5
Использование новых уникальных ассетов для глобальных иVENTов	4	0
Качество новых уникальных ассетов для глобальных иVENTов	0	6
<b>Итого</b>	<b>50</b>	

## **Блок Программирование (100 баллов)**

Таблица 4.3.5. Критерии оценивания

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<b>Задание 1. Пользовательский контент</b>	<b>35</b>	<b>15</b>
Количество возможных изменений: дополнения способны изменять минимум 3 основных внутриигровых части игры (уровни, предметы, персонажи, локации, квесты и т. д.). По два балла за каждую часть (максимум 6). <i>Пример:</i> <i>Возможность изменения внутриигровых диалогов.</i> <i>Изменение скорости появления квестов.</i>	6	0
Добавление своих файлов: <ul style="list-style-type: none"> <li>● пользователь может добавлять в дополнения свои файлы;</li> <li>● минимум три разные части, для которых юзер может использовать свои файлы.</li> </ul> По два балла за каждую часть (максимум шесть). <i>Примеры:</i> <i>Загрузка своих текстур к внутриигровым моделькам.</i> <i>Загрузка своих json файлов с характеристиками врагов.</i>	6	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Изменения большинства элементов: <ul style="list-style-type: none"> <li>● изменения и загрузка своих файлов применяемы для большинства сущностей данного (изменяемого) типа;</li> <li>● неединичная возможность применения изменений.</li> </ul> <p><i>Примеры:</i>  <i>Если у юзера есть возможность изменять UI в игре, это должен быть не один элемент, а большинство элементов UI.</i>  <i>По такому же принципу, изменения диалогов должны касаться не только одного нпс, а большинства возможных.</i></p>	7	3
Импорт UGC: реализована возможность импорта пользовательских дополнений в игру.	3	0
Экспорт UGC: реализована возможность экспорта созданных дополнений из игры.	3	0
Приложены примеры UGC: приложено не менее трех примеров пользовательского контента.	4	0
Разнообразие и качество примеров UGC: примеры UGC демонстрируют разнообразие и качество созданного контента, а также раскрывают все возможности вашей реализации создания пользовательских дополнений.	0	6
Внутренний контент руководства: руководство содержит от 3 000 символов до 10 000 и от пяти информативных скриншотов из игры, сопровождающих текст.	2	
Полнота руководства: руководство информативно, понятно и охватывает всю необходимую информацию для работы с пользовательским контентом, включая: импорт, экспорт, загрузку своих файлов и т. д., а также описывает все возможности создания UGC (какие изменения куда можно вносить, на что это повлияет).	0	2
Доступность руководства: руководство доступно в игровом интерфейсе или на внешнем ресурсе, ссылка на который указана в игре.	1	0
Воспроизводимость создания модов: проверяющий сумел, следуя приложенному руководству, сделать свой мод.	0	4
Предупреждения об ошибках: не возникло ошибок при попытке проверяющего, следуя руководству, сделать свой мод, или руководство предупреждает обо всех ошибках, которые могли возникнуть и возникли.	3	0
<b>Итого. Обязательное задание</b>	<b>50</b>	
<b>Задание 2. Процедурная генерация</b>	<b>45</b>	<b>5</b>
Параметры генерации: <ul style="list-style-type: none"> <li>● реализовано не менее 10 параметров (например: высота ландшафта, плотность врагов, типы ресурсов, климатические зоны, квесты);</li> <li>● каждый параметр влияет на разные аспекты генерации.</li> </ul>	10	0
Разнообразие и баланс: параметры обеспечивают как разнообразие (например, разные типы ландшафтов), так и баланс (например, равномерное распределение ресурсов).	2	2

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реализацию</b>	<b>экспертной оценки</b>
Ручной ввод сидов: ● реализована возможность генерации уровня по введенному юзером сиду; ● сид явно передается в алгоритм генерации и используется для инициализации вашей системы генерации.	5	0
Различная генерация при разных сидах: при последовательном вводе нескольких случайных сидов нет четкой взаимосвязи между генерациями.	5	0
Идентичное воспроизведение: при вводе одинакового сида на разных устройствах генерируются идентичные ключевые элементы (например, расположение гор, врагов, квестов).	5	0
Подстраивание под уровень игрока: учитываются минимум три параметра игрока для дальнейшего изменения сложности генерации, к примеру: ● уровень; ● время прохождения; ● количество смертей; ● пройденные квесты; ● собранные ресурсы.	3	0
Динамическое изменение: сложность генерации изменяется от одного прохождения к другому (прохождением в данном случае называется цикл от одной генерации к следующей). <i>Примеры:</i> <i>Плотность врагов больше, сложность квестов выше после успешных забегов игрока.</i> <i>Больше лечебных (или других вспомогательных) предметов, враги слабее при частых смертях или неудачных выполнениях квестов.</i>	7	0
Система избегает крайностей: ● есть пороговые значения (например, враги не могут исчезнуть полностью); ● изменения плавные (не скачкообразные).	4	0
Блок-схема генерации уровня: предоставлена блок-схема, созданная на сайте draw.io с форматом файла .drawio, охватывающая основные этапы системы генерации.	1	0
Логичность этапов: четко показана последовательность шагов и связи между ними (например, как сид влияет на каждый этап).	0	3
Стандартизация блок-схемы: использованы общепринятые стандартные символы (ромбы для решений, прямоугольники для процессов и т. д.).	1	0
Детализация: каждый этап имеет пояснение (например, «генерация высот на основе шума Перлина»), нет неоднозначных элементов.	2	0
<b>Итого. Задание на выбор</b>	<b>50</b>	
<b>Задание 3. Глобальные ивенты</b>	<b>45</b>	<b>5</b>

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Хранение ивентов на сервере: глобальные ивенты хранятся на сервере и были добавлены, используя «Запрос на создание Ивента» из инструкции*.	1	0
Понятность названий ивентов: названия ивентов (поле name) (полученных запросом из инструкции* с предоставленного сервера) легко читаются и могут быть интуитивно сопоставлены с реализованными ивентами внутри игры.	0	2
Частота запросов на сервер: игра запрашивает ивенты с предоставленного организаторами сервера не чаще одного раза в 2 мин.	3	0
Обработка ошибок при запросах: <ul style="list-style-type: none"> <li>● реализована обработка ошибок при запросе к предоставленному организаторами серверу (например, отсутствие интернета);</li> <li>● событие все равно произойдет, используя старые данные предыдущих запросов.</li> </ul>	4	0
Уведомления об ивентах: игрок получает уведомления о скором начале события (полученные запросом из инструкции* с предоставленного сервера): всплывающие окна, иконки на карте, звуковые сигналы (что-то одно из этого обязательно, остальное по желанию).	3	0
Настройка уведомлений: уведомления о предстоящих событиях (полученных запросом из инструкции* с предоставленного сервера) должны быть настраиваемыми внутри игры (например, за 1 день, за 15 мин), а также могут быть полностью отключены.	3	0
Одновременное воспроизведение: одни и те же ивенты (полученные запросом из инструкции* с предоставленного сервера) запускаются в одно время на разных устройствах, даже на разных часовых поясах (максимальная разница — 2 мин).	2	0
Динамическое изменение: ивенты (полученные запросом из инструкции* с предоставленного сервера) динамически изменяются, подстраиваясь под изменение данных на сервере. (К примеру, если на сервере была изменена длительность ивента duration_in_minutes, то ивент должен продлиться дольше, соответствуя новому значению, или, если ивент был полностью удален, то и в игре он должен закончиться.)	10	0
Геймплейные изменения: ивенты (полученные запросом из инструкции* с предоставленного сервера) влияют на геймплей. Не менее трех изменений в геймплее для каждого ивента (изменением может быть не только новая механика, но и измененная старая механика или результат ее действия).	5	0
Визуальные изменения: ивенты (полученные запросом из инструкции* с предоставленного сервера) визуально заметны в UI; должно быть не менее трех новых (или измененных) элементов UI для каждого ивента.	5	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Структурность ивентов: ивенты (полученные запросом из инструкции* с предоставленного сервера) не конфликтуют друг с другом (влияют на разные части игры, если происходят в одно время).	5	0
Блок-схема системы ивентов: предоставлена блок-схема, созданная на сайте draw.io** с форматом файла .drawio, охватывающая основные этапы системы ивентов.	1	0
Логичность этапов: четко показана последовательность шагов и связи между ними (например, как игра узнает, когда уведомить игрока о наступающем ивенте).	0	3
Стандартизация блок-схемы: использованы общепринятые стандартные символы (ромбы для решений, прямоугольники для процессов и т. д.).	1	0
Детализация: каждый этап имеет пояснение (например, «получение списка ивентов с сервера»), нет неоднозначных элементов.	2	0
<b>Итого. Задание на выбор</b>	<b>50</b>	

\* <https://disk.yandex.ru/i/MD88qfhWUusJSA>.

\*\* <http://draw.io/>.

## **Блок Продукт (150 баллов)**

Таблица 4.3.6. Критерии оценивания

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<b>Часть 1. Страница</b>		
Заполнение и единобразие страницы, отсутствие ошибок в тексте	3	2
Логотип игры	1	0
Баннер игры	1	1
Наличие краткого описания (до 40 слов)	1	1
Наличие полного описания (от 100 до 250 слов)	1	1
Скриншоты из игры 3/6, имеют одно разрешение	2	3
Дорожная карта развития проекта (более 5 пунктов)	1	2
<b>Итого</b>	<b>20</b>	
<b>Часть 2. Трейлер</b>		
Техника монтажа	5	5
Режиссура	0	10
Присутствует название команды и игры	2	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Использованы кадры реального игрового процесса и постановочные кадры	4	0
Понятен жанр игры и игровые возможности	4	0
<b>Итого</b>	<b>30</b>	
<b>Часть 3. Демоверсия игры</b>		
Игра является immersive (погружающей). (Насколько игра захватывает игрока, создает ощущение присутствия в игровом мире.)	5	0
Качественно, разнообразно и сбалансировано реализован основной игровой процесс (Core-геймплей). (Глубина механик, баланс сложности и взаимодействие игровых систем.)	5	0
Продуман и увлекательен мета-геймплей (прогресс персонажа, прокачка, коллекционирование и т. д.). (Глубина систем прогрессирования и их влияние на мотивацию игрока.)	5	0
Хорошо проработана игровая среда и мир. (Детализация игрового мира, его интерактивность, атмосфера и то, насколько такой мир способствует погружению игрока.)	5	0
Присутствие в игре элементов, побуждающих к повторному прохождению. (Наличие альтернативных концовок, случайной генерации, дополнительных режимов и других факторов, повышающих реиграбельность.)	5	0
Стабильность работы игры. (Отсутствие критичных багов, вылетов и проблем с производительностью на всех платформах.)	5	0
Игровой интерфейс качественно продуман и удобен. (Интуитивность, функциональность и удобство навигации.)	5	0
Хорошо реализована система управления. (Отзывчивость, удобство и адаптируемость управления под разные устройства.)	5	0
Качественно реализована графика и визуальные эффекты. (Детализация, художественный стиль, анимация и общее визуальное впечатление.)	5	0
Качественно проработаны звуковые и музыкальные эффекты. (Соответствие звукового оформления атмосфере и геймплею игры.)	5	0
Игра хорошо обучает игрока. (Наличие и качество обучающих элементов, которые помогают понять правила и механики.)	5	0
Реализованность системы наград и мотивации. (Система наград (достижения, внутриигровые бонусы, награды за выполнение задач) стимулирует игрока продолжать играть.)	5	0
<b>Итого</b>	<b>60</b>	
<b>Часть 4. Питч</b>		
Четкость и структурированность описания геймплея. (Ясно и логично изложены основные механики и особенности игрового процесса?)	5	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
Оригинальность и привлекательность идеи игры. (Насколько уникальна и интересна концепция игры? Есть ли в ней элементы, которые выделяют ее на фоне конкурентов?)	5	0
Глубина анализа конкурентов. (Подробно и объективно проведен анализ конкурентов. Учтены их сильные и слабые стороны.)	5	0
Точность анализа целевой аудитории. (Насколько точно определена целевая аудитория? Учтены ли ее предпочтения, возрастные и поведенческие характеристики?)	5	0
Обоснованность выбора стиля игры. (Насколько логично выбранный стиль соответствует концепции игры и ожиданиям целевой аудитории?)	5	0
Уникальность и инновационность игры. (Насколько игра предлагает новые или улучшенные решения в сравнении с существующими проектами?)	5	0
Реалистичность и масштабируемость проекта. (Насколько реалистичны планы по разработке и масштабированию игры? Учтены ли технические и ресурсные ограничения?)	5	0
Качество обоснования ответов на вопросы. (Насколько аргументированно и уверенно даны ответы на вопросы? Подкреплены ли они фактами и данными?)	5	0
<b>Итого</b>	<b>40</b>	

## **Блок Берлога (50 баллов)**

Таблица 4.3.7. Критерии оценивания

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реали- зацию</b>	<b>экспертной оценки</b>
<b>Задание 1. Игровой таймер*</b>		
Интерактивное обучение основам управления	2	0
Постепенное введение обучающих механик игры	2	0
Визуальные подсказки и указатели	2	0
Возможность повторного прохождения обучения	2	0
Обратная связь за успехи в обучении	2	0
Ясность и понятность инструкций в обучении	0	5
Уместность и контекстуальность обучения (обучение происходит в подходящий момент)	0	5
Наличие продвинутых техник обучения, интересных и запоминающихся	0	5
<b>Итого</b>	<b>25</b>	
<b>Задание 2. Интро*</b>		
Соответствие техническим требованиям к интро	2	0

<b>Критерии</b>	<b>Максимальные баллы</b>	
	<b>за реализацию</b>	<b>экспертной оценки</b>
Соответствие стилю и атмосфере игры	2	0
Интро вовлекает игрока в сюжет игры	2	0
Интро углубляет образы ключевых персонажей игры, вскрывает их мотивации	2	0
Интро обозначает основной конфликт/проблему игры	2	0
Качество художественной реализации (в случае анимации или иллюстраций — детализация, узнаваемость персонажей; в случае акцента на аудио — качество звуковых эффектов; в случае текста — стилистические приемы)	0	5
Удобство восприятия. Интро не затянуто, но дает достаточно информации, тексты (при наличии) читаемы	0	5
Соответствие сеттингу Берлога	0	5
<b>Итого</b>	<b>25</b>	

\* <https://disk.yandex.ru/i/o1dFEcSu1h6usg>.

### 4.3.7. Материалы для подготовки

#### *Программирование*

1. Курс по основам языка программирования C# : [https://ulearn.me/Course/BasicProgramming2/Steki\\_i\\_ocheredi\\_48016626-87ae-411d-ae97-f7a49e465dbc](https://ulearn.me/Course/BasicProgramming2/Steki_i_ocheredi_48016626-87ae-411d-ae97-f7a49e465dbc).
2. Открытый курс от Яндекс по разработке игр на Unity: <https://www.youtube.com/playlist?list=PLxAJaawE-vksJlTcZKibYEmtL4dUqbrSa>.
3. Канал Brackeys на тему «Разработка игр»: <https://www.youtube.com/@Brackeys/videos>.
4. Канал «ARTALASKY CG – Как создать игру»: <https://www.youtube.com/@DenOvsyannikov/videos>.
5. Документация Unity с примерами кода: <https://docs.unity3d.com/Manual/index.html>.

#### *3D-графика*

1. Открытый курс от Kaino по 3D-моделированию: <https://online.kaino.ru/training/view/BlenderStart>.
2. Открытый курс от Kaino по 3D-моделированию: <https://online.kaino.ru/training/view/Wagon>.
3. Открытый уроки по Blender: <https://www.youtube.com/playlist?list=PLrsq-o51mMFHlysQmjQTkkPyTMMGi2-Pw>.
4. Полезные видеоролики по 3D-моделированию от Denis Kozhar: <https://www.youtube.com/@DenisKozhar/videos>.

5. Полезные видеоролики по 3D-моделированию от Danil Gryzlov: <https://www.youtube.com/@dgryzlovart/videos>.

## **2D-графика**

1. Открытый курс «Фотошоп»: [https://www.youtube.com/watch?v=dYQBuW8rBbI&list=PL\\_jKgaFUK\\_61plyXULw7KPi6HGpyXKTWx&index=4](https://www.youtube.com/watch?v=dYQBuW8rBbI&list=PL_jKgaFUK_61plyXULw7KPi6HGpyXKTWx&index=4).
2. Открытый курс Figma: <https://www.youtube.com/watch?v=SWBUS99H1fc&list=PLrsq-o51mMFGdWoz-ETc4fTgzVJKPrEzI>.
3. Канал ANIMATION CLUB посвященный анимации и режиссуре: [https://www.youtube.com/@animation\\_club/videos](https://www.youtube.com/@animation_club/videos).

## **Геймдизайн**

1. Книга по основам геймдизайна: «Геймдизайн», Джесси Шелл (неполная версия): <https://www.rulit.me/books/gejmdizajn-kak-sozdat-igru-v-kotoruyu-budut-igrat-vse-litres-read-588467-14.html>.
2. Книга о разработке компьютерных игр: «Кровь, пот и пиксели», Джейсон Шрейер (неполная версия): <https://www.rulit.me/books/krov-pot-i-pikseli-obratnaya-storona-industrii-videoigr-read-525334-1.html>.
3. Открытый курс по геймдизайну: <https://edvice.pro/online-course/game-design/>.
4. Открытый курс по lvl-дизайну: <https://edvice.pro/online-course/level-design/>.

# **5. Критерии определения победителей и призеров**

## **Первый отборочный этап**

В первом отборочном этапе участники решали задачи предметного тура по двум предметам: математике и информатике и инженерного тура. В каждом предмете максимально можно было набрать 100 баллов, в инженерном туре 100 баллов. Для того чтобы пройти во второй этап, участники должны были набрать в сумме по обоим предметам и инженерному туре не менее 50,0 баллов, независимо от уровня.

## **Второй отборочный этап**

Количество баллов, набранных при решении всех задач второго отборочного этапа, суммируется. Победители второго отборочного этапа должны были набрать не менее 400,0 баллов, независимо от уровня.

## **Заключительный этап**

### ***Индивидуальный предметный тур***

- математика — максимально возможный балл за все задачи — 100 баллов;
- информатика — максимально возможный балл за все задачи — 100 баллов.

### ***Командный инженерный тур***

Команды заключительного этапа получали за командный инженерный тур от 0 до 460,00 баллов: команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем.

Все результаты команд нормировались по формуле:

$$\frac{100 \times x}{MAX},$$

где  $x$  — число баллов, набранных командой,

$MAX$  — число баллов, максимально возможное за инженерный тур.

В заключительном этапе олимпиады индивидуальные баллы участника складываются из двух частей, каждая из которых имеет собственный вес: баллы за индивидуальное решение задач по предмету 1 (математика) с весом  $K_1 = 0,15$ , по

предмету 2 (информатика) с весом  $K_2 = 0,15$ , баллы за командное решение задач инженерного тура с весом  $K_3 = 0,7$ .

Итоговый балл определяется по формуле:

$$S = K_1 \cdot S_1 + K_2 \cdot S_2 + K_3 \cdot S_3,$$

где  $S_1$  — балл первой части заключительного этапа по математике (предметный тур) ( $S_{1\text{ макс}} = 100$ );

$S_2$  — балл первой части заключительного этапа по информатике (предметный тур) ( $S_{2\text{ макс}} = 100$ );

$S_3$  — итоговый балл инженерного командного тура ( $S_{3\text{ макс}} = 100$ ).

Итого максимально возможный индивидуальный балл участника заключительного этапа — 100 баллов.

### ***Критерий определения победителей и призеров***

Чтобы определить победителей и призеров (независимо от класса) на основе индивидуальных результатов участников, был сформирован общий рейтинг всех участников заключительного этапа. С начала рейтинга были выбраны 4 победителя и 11 призеров (первые 25% участников рейтинга становятся победителями или призерами, из них первые 8% становятся победителями, оставшиеся — призерами).

### ***Критерий определения победителей и призеров (независимо от уровня)***

Категория	Количество баллов
Победители	57,58 и выше
Призеры	От 47,62 до 57,08

## **6. Работа наставника после НТО**

Участие школьника в Олимпиаде может завершиться после любого из этапов: первого или второго отборочных, либо после заключительного этапа. В каждом случае после завершения участия наставнику необходимо провести с учениками рефлексию — обсудить полученный опыт и проанализировать, что позволило достичь успеха, а что привело к неудаче. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: <https://academy.sk.ru/events/310>.

Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (Р2Р). Они могут выявить внутренние проблемы команды, для решения которых в план подготовки можно добавить мероприятия, направленные на ее сплочение.

Стоит рассказать, что в истории НТО было много примеров, когда не победив в первый раз, на следующий год участники показывали впечатляющие результаты, одержав победу сразу в нескольких профилях. Конечно, важно отметить, что так происходит только при учете прошлых ошибок и подготовке к Олимпиаде в течение года.

Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с ними помогает наставнику продемонстрировать важность компетенций, развиваемых в процессе участия в НТО, для будущего образования и карьеры школьников. Поддержка родителей помогает мотивировать участников и позволяет выделить необходимое время на занятия в кружке.

С участниками-выпускниками наставнику рекомендуется обсудить их дальнейшее профессиональное развитие и его связь с выбранными профилями НТО. Отдельно можно обратить внимание на льготы для победителей и призеров, предлагаемые в вузах с интересующими ученика направлениями. Кроме того, ряд вузов предлагает льготы для всех финалистов НТО, а также учитывает результаты Конкурса цифровых портфолио «Талант НТО».