МАТЕРИАЛЫ ЗАДАНИЙ
Всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада»

по профилю «Разработка мобильных приложений»

2024/25 учебный год

ntcontest.ru

УДК 373.5.016:[794:004.946]
ББК 74.263.2
Р17

Авторы:
И. А. Воронцов, А. А. Гаврилюк, Е. Н. Горечин, О. В. Зубков, А. М. Лимасов, А. А. Маркин, А. Е. Непретимов, А. И. Таразевич, И. В. Ширстова, В. К. Шперлинг

Р17 Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада». Учебно-методическое пособие
Том 28 **Разработка мобильных приложений**
— М.: Ассоциация участников технологических кружков, 2025. — 185 с.

ISBN 978-5-908021-27-2

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

УДК 373.5.016:[794:004.946]
ББК 74.263.2

Оглавление

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 робототехническими соревнованиями;
- школьников, увлеченных олимпиадами и предпочитающих межпредметный подход;
- энтузиастов передовых технологий;
- активных участников хакатонов, проектных конкурсов и профильных школ;
- будущих предпринимателей, ищущих 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: помощь 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 Кружко-

вое 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 помощь 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 мобильных приложений

Цель профиля Разработка мобильных приложений — формирование у школьников прикладных навыков и компетенций, необходимых для создания мобильных приложений полного цикла: от проектирования интерфейса и клиентской логики до построения серверной архитектуры и взаимодействия через API. Олимпиада позволяет участникам погрузиться в реальные сценарии работы в IT-компании, пройти путь от собеседования до выполнения проектного задания в условиях командной разработки.

Концепция профиля основана на деловой игре «Стажировка в компании S-Mobile». Все этапы Олимпиады оформлены как симуляция поступления и прохождения стажировки в разработческой команде мобильного стартапа. Это вовлекает участников в контекст прикладных технологических задач и позволяет на практике применять знания в области Android-разработки, UI/UX-дизайна и бэкенда.

Первый отборочный этап включает инженерный тур в виде теста, где оцениваются навыки анализа информации, чтения документации, работы с SQL, JSON и REST API. Здесь проверяется базовая IT-грамотность и способность к самостоятельному обучению.

На втором отборочном этапе участники переходят к выполнению практических задач в выбранной роли: Android-разработчика, UI/UX-дизайнера или бэкенд-разработчика. Индивидуальные задания позволяют им развить профильные навыки, а командные — попробовать себя в совместной работе, спланировать и реализовать прототип реального мобильного приложения.

Инженерный тур заключительного этапа ставит задачу разработки мобильного приложения и серверной части в рамках кейса, приближенного к задачам реальной индустрии. Участники создают приложение для системы СКУД, обеспечивающее вход по QR-коду и предоставляющее администраторам возможность управления пропусками. Особое внимание уделяется вопросам безопасности, архитектуры, тестирования, а также качеству UX-дизайна. Здесь, в том числе, делается акцент на оценивание индивидуального вклада каждого в решение финальной командной задачи.

Каждый этап последовательно вовлекает участников в решение все более комплексных задач — от индивидуальных до командных, — развивая не только технические, но и проектные, коммуникационные и управленческие компетенции.

Профиль востребован среди школьников, интересующихся мобильной разработкой, и может стать отличной стартовой площадкой для поступления в ведущие вузы по направлениям «Прикладная информатика», «Программная инженерия», «Информационные технологии». Выпускники профиля успешно применяют полученные навыки в университетских проектах, хакатонах и первых стажировках в IT-компаниях.

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 Контест. 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 $0 \leqslant t \leqslant 10^6$.

Формат выходных данных

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

Примеры

Пример №1

Стандартный ввод
6
Стандартный вывод
120

Пример №2

Стандартный ввод
2
Стандартный вывод
0

Пример №3

Стандартный ввод
1000000
Стандартный вывод
999997000002000000

Решение

Ниже представлено решение на языке C++.

C++

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

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 $1 \leqslant a, b, c, d \leqslant 10^9$.

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

Стандартный ввод
5 10 9 6
Стандартный вывод
5

Пример №2

Стандартный ввод
4 10 9 6
Стандартный вывод
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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    double a, b, c, d;
    cin >> a >> b >> c >> d;
    double a0 = min({a, b, c, d});
    double a1 = min(max(a, b) / 2.0, min(a, b));
    double a2 = min(max(c, d) / 2.0, min(c, d));
    double ans =  max({a0, a1, a2});
    if( (int)ans == ans ){
        int ians = ans;
        cout << ians << endl;
        return 0;
    }
    cout.precision(1);
    cout << fixed<< ans << endl;
}
```

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, старик забыл, какова была исходная длина целых досок.

Формат входных данных

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

Во второй строке заданы $2n$ целых чисел d_i — длины всех кусков, которые получились после «тренировки» внука, где $1 \leqslant d_i \leqslant 10^9$. 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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    vector<pair<int, int> > v(2 * n);
     for(int i = 0; i < 2 * n; i++){
        int d;
        cin >> d;
        v[i] = {d, i + 1};
    }
    sort(v.begin(), v.end());
    vector<pair<int, int> > ans(n);
    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 $2 \leqslant$ $\leqslant a \leqslant 10^{18}$, $1 \leqslant b$, $c \leqslant 10^{18}$.

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

Стандартный ввод
4 5 41
Стандартный вывод
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 каждых 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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
int f1(int M, int a){
    int res = 0, z = 0;
    while(1){
        if(M == 0 && z < a){
            return res;
        }
        res += M;
        M = M + z;
        z = M % a;
        M = M / a;
    }
}
```

```
int f2(int M, int b){
    return M + M / b;
}
signed main(){
    int a, b, c;
    cin >> a >> b >> c;
    int L = 0, R = 1;
    while(f1(R, a) <= c){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f1(M, a) < c){
            L = M;
        }
        else{
            R = M;
        }
    }
    int z = R;
    L = 0, R = 1;
    while(f2(R, b) <= z){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f2(M, b) < z){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << R << endl;
}
```

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, заработок 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 $2 \leqslant n, m \leqslant 2 \cdot \cdot 10^5$.

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. $1 \leqslant A, B \leqslant n$, $1 \leqslant d \leqslant 10^9$. 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, $1 \leqslant S \leqslant \leqslant n$.

Формат выходных данных

Вывести 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

Стандартный ввод
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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
int n, m;
vector<vector<pair<int, int> > > G;
vector<int> order, used;
void dfs(int a){
    used[a] = 1;
    for(auto to : G[a]){
        if(!used[to.first]){
            dfs(to.first);
        }
    }
    order.push_back(a);
}
signed main(){
    cin >> n >> m;
    G.resize(n + 1);
    used.resize(n + 1, 0);
    for(int i = 0; i < m; i++){
        int a, b, d;
        cin >> a >> b >> d;
        G[a].push_back({b, d});
    }
```

```
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 $n \times m$.

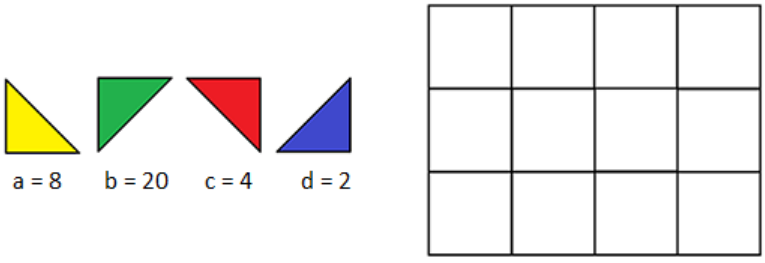


Рис. 2.2.1

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

Стандартный ввод
`8 20 4 2` `3 4`

Стандартный вывод
`18`

Примечания

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

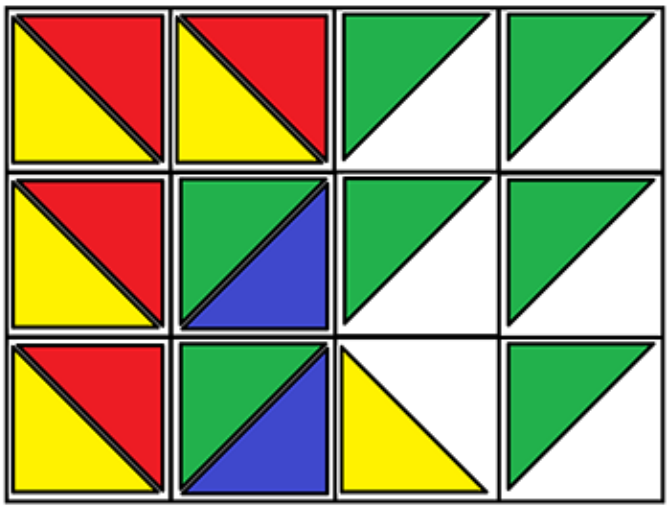


Рис. 2.2.2

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d, n, m;
    cin >> a >> b >> c >> d >> n >> m;
    if(a > c){
        swap(a, c);
    }
    if(b > d){
        swap(b, d);
    }
    int f = a + b;
    int k = n * m;
    if(k <= f){
        cout << k * 2;
        return 0;
    }
    k -= f;
    c -= a;
    d -= b;
    cout << f * 2 + min(k, c + d) << endl;
}
```

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 $1 \leqslant n \cdot k \leqslant 10^{18}$.

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

Стандартный ввод
`45 2`

Стандартный вывод
`8 12 5 20`

Пример №2

Стандартный ввод
`128 7`

Стандартный вывод
`7 21 2 98`

Решение

Ниже представлено решение на языке C++.

C++

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

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, $2 \leqslant n, m \leqslant 25$. Во второй строке через пробел находятся координаты (x, y) обязательной для посещения клетки, где $1 \leqslant x \leqslant n$, $1 \leqslant y \leqslant m$. 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 3×4, обязательно посещая по пути клетку $(2, 3)$.

Комментарий

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

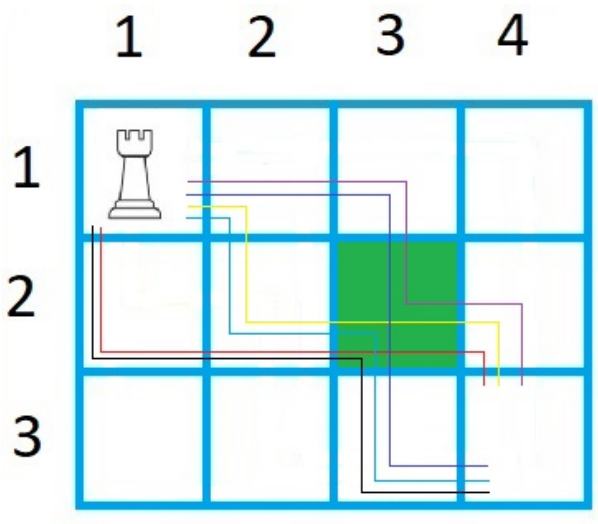


Рис. 2.2.3

Решение

Ниже представлено решение на языке C++.

C++

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

```
            bc[i][j] += bc[i - 1][j];
            if(j - 1 >= 0){
                bc[i][j] += bc[i - 1][j - 1];
            }
        }
    }
    int n, m, x, y;
    cin >> n >> m >> x >> y;
    int d1 = bc[x - 1 + y - 1][x - 1];
    int d2 = bc[n - x + m - y][n - x];
    int ans = d1 * d2;
    cout << ans << endl;
}
```

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 футболку, 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 скуки 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

Стандартный ввод
1456531355
Стандартный вывод
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 левых 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++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    string s;
    cin >> s;
    string mx = s, mn = s;

    for(int i = 0; i < 5; i++){
        for(int j = 0; j < 10; j++){
            mx = max(mx, s);
            mn = min(s, mn);
            if(j < 9){
                s = s.substr(1) + s[0];
            }
        }
        for(int j = 0; j < 5; j++){
            swap(s[2 * j], s[2 * j + 1]);
        }
    }
    stringstream ssmn;
    ssmn << mn;
    int imn;
    ssmn >> imn;
    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 $1, \ldots, n$.

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 $1 \leqslant \leqslant n \leqslant 2 \cdot 10^5$.

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

Стандартный ввод
9 2 9 1 5 6 4 3 8 7
Стандартный вывод
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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int LG = 19;
int N = (1 << LG);
vector<int> tr(2 * N, 0);
void upd(int pos, int x){
    pos += N;
    tr[pos] = x;
    pos /= 2;
    while(pos){
        tr[pos] = {tr[2 * pos]+ tr[2 * pos + 1]};
        pos /= 2;
    }
}
int get(int l, int r){
    l += N;
    r += N;
    int res = 0;
    while(l <= r){
        if(l % 2 == 1){
            res += tr[l];
        }
        if(r % 2 == 0){
            res += tr[r];
        }
        l = (l + 1) / 2;
        r = (r - 1) / 2;
    }
    return res;
}
signed main(){
    int n, a;
    cin >> n;
    int ans = 0;
    for(int i = 0; i < n; i++){
        cin >> a;
        int sl = get(0, a - 1);
        int sr = get(a + 1, N - 1);
        ans += min(sl, sr);
        upd(a, a);
    }
    cout << ans << endl;
}
```

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

Стандартный ввод
`1 5 10`
Стандартный вывод
`3`

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int A, C, D;
    cin >> A >> C >> D;
    cout.precision(1);
    cout << fixed << (A + C) / 2.0 << endl;
}
```

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

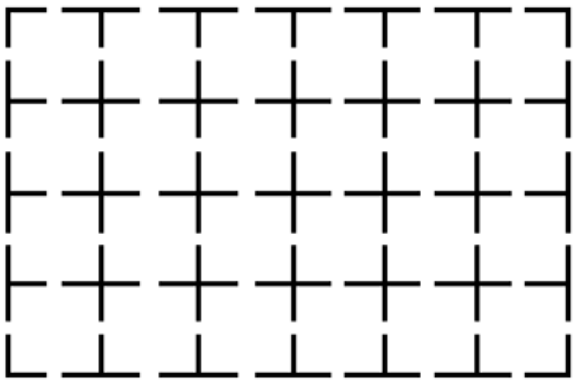


Рис. 2.2.4

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 $1 \leqslant n,\ m \leqslant 10^9$.

Формат выходных данных

Требуется по числам a и b найти размеры исходной решетки n и m. Будем всегда считать, что $n \leqslant m$, 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++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b;
    cin >> a >> b;
    int L = 0, R = a / 4 + 1;
    while(R - L > 1){
        int M = (R + L) / 2;
        int D = a / 2 - M;
        if(M * D <= b){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << L + 1 << ' ' << a / 2 - L + 1 << endl;
}
```

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 учти, 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 $2 \leqslant n,\ m \leqslant 1\,000$.

Во второй строке находятся два числа x и y через пробел — координаты свечки, где $1 \leqslant x \leqslant n - 1$, $1 \leqslant y \leqslant m - 1$, 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

Стандартный ввод
8 5 7 2
Стандартный вывод
12.571

Пример №2

Стандартный ввод
2 2 1 1
Стандартный вывод
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.

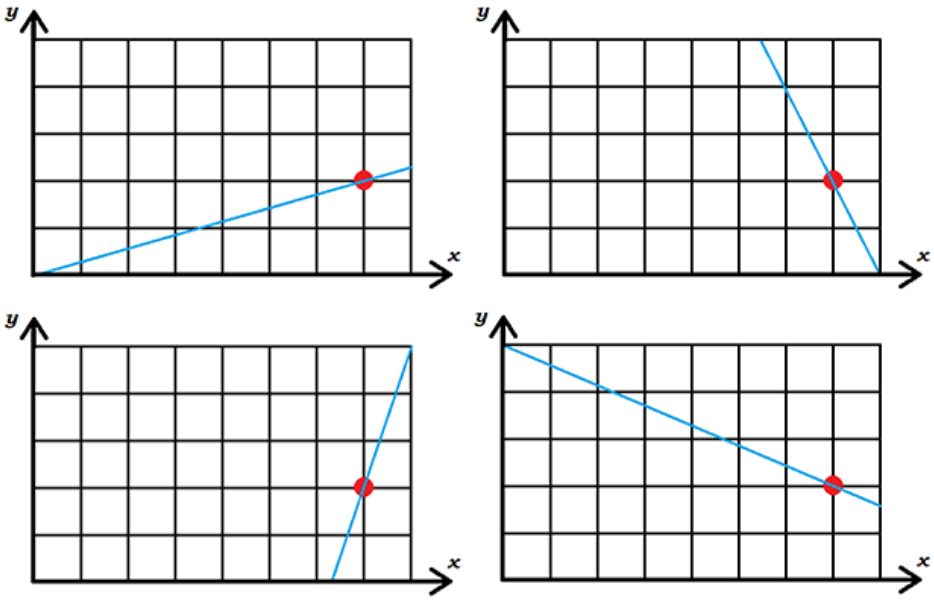


Рис. 2.2.5

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++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
double katy(double x, double y, double n){
    return n * y / x;
}
double n, m, x, y;
double ans = INF;
double k1, k2;
void upd(){
    if(k1 < m){
        double st =k1 * n / 2;
        ans = min(ans, n * m - 2 * st);
    }
    else{
        double st =k2 * m / 2;
        ans = min(ans, n * m - 2 * st);
    }}
signed main(){
    cin >> n >> m >> x >> y;
    k1 = katy(x, y, n);
    k2 = katy(y, x, m);
    upd();
    k1 = katy(n - x, y, n);
    k2 = katy(y, n - x, m);
    upd();
    k1 = katy(x, m - y, n);
    k2 = katy(m - y, x, m);
    upd();
    k1 = katy(n - x, m - y, n);
    k2 = katy(m - y, n - x, m);
    upd();
    cout.precision(3);
    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 $2 \leqslant n \leqslant 2 \cdot 10^5$. Во второй строке через пробел заданы n целых чисел a_i через пробел — длины полосок, где $1 \leqslant a_i \leqslant 10^9$.

Формат выходных данных

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

Примеры

Пример №1

Стандартный ввод
8 3 6 5 4 4 5 5 2

Стандартный вывод
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 3×4, и ее площадь 12. При ее создании Маша не должна использовать полоску номер 8, по этой причине неважно, как она раскрашена.

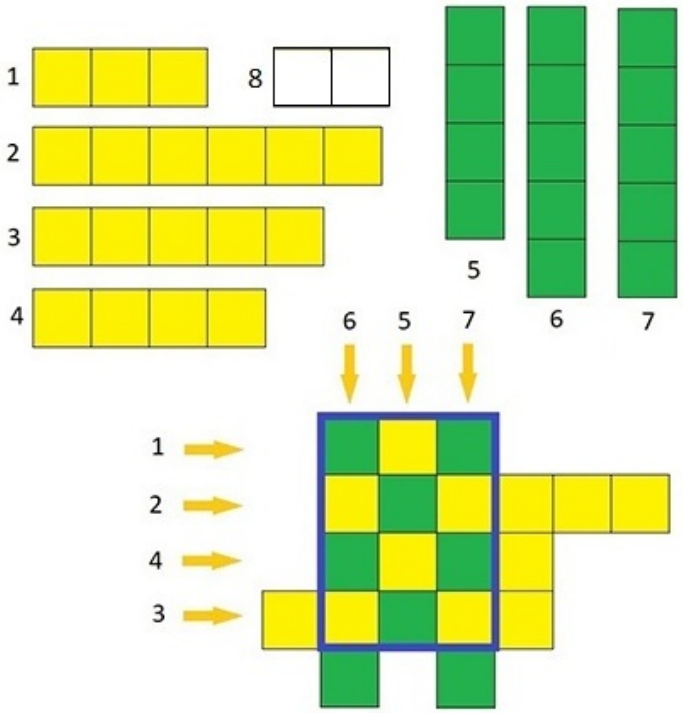


Рис. 2.2.6

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    deque<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    sort(v.begin(), v.end());
    int ans = 0;
    int cnth = 0, minh;
    while(1){
        if(v.size() == 0){
            break;
        }
        cnth++;
        minh = v.back();
        v.pop_back();
        while(v.size() > 0 && v[0] < cnth){
            v.pop_front();
        }
        ans = max(ans, cnth * min(minh, (int)v.size()));
    }
    cout << ans << endl;
}
```

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 $20 \leqslant n \leqslant 100$, 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, $20 \leqslant$ $\leqslant n \leqslant 100$.

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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
string frs;
int n;
vector<string> dict;
vector<int> msk(26, 0);
int cnt = 0;
vector<vector<int> > amsk;
vector<string> ans;
bool bigok = 0;
void p(int pos){
   if(!bigok){
        if(cnt == 0){
            sort(ans.begin(), ans.end());
            bigok = 1;
            return;
        }
        for(int i = pos; i < n; i++){
            string ts = dict[i];
            bool ok = 1;
            for(int j = 0; j < 26; j++){
                if(amsk[i][j] > msk[j]){
                    ok = 0;
                }
            }
            if(ok){
                ans.push_back(ts);
                for(int j = 0; j < 26; j++){
                    msk[j] -= amsk[i][j];
                    cnt -= amsk[i][j];
                }
                p(i + 1);
                if(!bigok){
                for(int j = 0; j < 26; j++){
                    msk[j] += amsk[i][j];
                    cnt += amsk[i][j];
                }
                ans.pop_back();
                }
            }
        }
   }
}
signed main(){
    cin >> frs;
    cin >> n;
    amsk.resize(n, vector<int>(26, 0));

    string ts;
    for(int i = 0; i < n; i++){
        cin >> ts;
        dict.push_back(ts);
    }
    for(int i = 0; i < n; i++){
        for(auto el : dict[i]){
            amsk[i][el - 'a']++;
        }
    }
```

```
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 $1 \leqslant a, b, c, d \leqslant 10^9$.

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

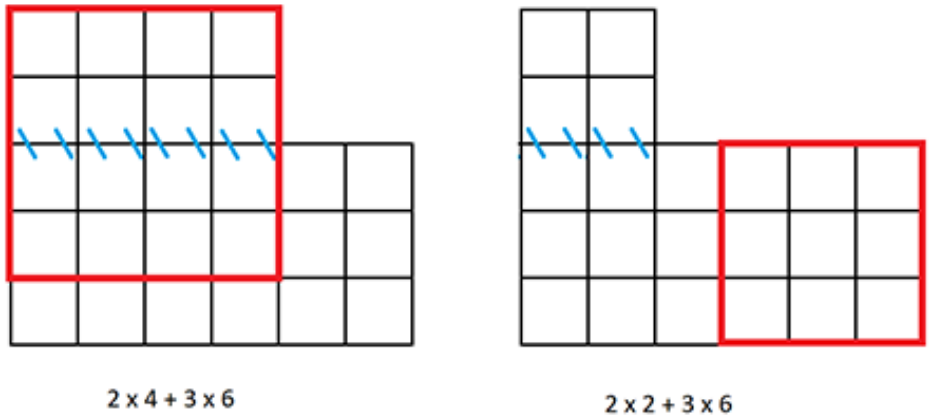


Рис. 2.2.7

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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    int ans = max(min(a, b), min(c, d));
    int p1 = min(a + c, min(b, d));
    int p2 = min(a + d, min(b, c));
    int p3 = min(b + c, min(a, d));
    int p4 = min(b + d, min(a, c));
    ans = max({ans, p1, p2, p3, p4});
    cout << ans << endl;
}
```

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 A, C, G, T.

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 префиксу длины i.

Например, пусть $D_1 =$ **AACGGTAGGT**, а $D_2 =$ TCCCGGAACA. Выберем $i = 4$ и поменяем местами префиксы длины 4. Получим две новые ДНК, одна из которых будет иметь вид **AACG**GGAACA, а вторая — TCCC**GTAGGT**. Для наглядности были выделены части первой из них.

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

Теперь можно рассмотреть популяцию из нескольких ДНК. Выберем из них две, произведем их скрещивание по префиксу 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 m ДНК, каждая имеет одну и ту же длину n. После некоторого количества попарных 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 $m-1$ 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 $10 \leqslant n \leqslant 100$, $2 \leqslant m \leqslant 100$.

В следующих m строках содержится описание исходной популяции ДНК, каждая задается строкой длины n, состоящей из символов A, C, G и T.

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
AATTCCTGAA

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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n, m;
    cin >> n >> m;
    vector<string> v1(m);
    for(int i = 0; i < m; i++){
        cin >> v1[i];
    }
    string d;
    cin >> d;
    vector<string> v2(m - 1);
    for(int i = 0; i < m - 1; i++){
        cin >> v2[i];
    }
    for(int j = 0; j < n; j++){
        int ss = 0;
        for(int i = 0; i < m; i++){
            ss ^= (int)(v1[i][j]);
        }
        for(int i = 0; i < m - 1; i++){
            ss ^= (int)(v2[i][j]);
        }
        cout << (char)(ss);
    }
    cout << endl;
}
```

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 временам 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 $1 \leqslant A, B, C, D \leqslant 1\,000$, $0 \leqslant E \leqslant 1\,000$.

Формат выходных данных

Вывести одно число — минимальное время прохождения туннеля всей группой.

Примеры

Пример №1

Стандартный ввод
8 9 10 1 3
Стандартный вывод
44

Пример №2

Стандартный ввод
8 9 10 1 0
Стандартный вывод
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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
vector<int> v(4);
int e, ans = INF;
void p(vector<int> &vl, vector<int> &vr, int tv){
    if(vl.size() == 2){
        ans = min(ans, tv + *max_element(vl.begin(), vl.end()));
        return;
    }
    for(int i = 0; i < vl.size() - 1; i++){
        for(int j = i + 1; j < vl.size(); j++){
            vector<int> vl1;
            for(int k = 0; k < vl.size(); k++){
                if(k != i && k != j){
                    vl1.push_back(vl[k]);
                }
            }
            vector<int> vr1 = vr;
```

```
            vr1.push_back(vl[i] + e);
            vr1.push_back(vl[j] + e);
            int tmp = max(vl[i], vl[j]);
            sort(vr1.rbegin(), vr1.rend());
            vl1.push_back(vr1.back() + e);
            vr1.pop_back();
            p(vl1, vr1, tv + tmp + vl1.back() - e);
        }
    }
}
signed main(){
    for(int i = 0; i < 4; i++){
        cin >> v[i];
    }
    sort(v.begin(), v.end());
    cin >> e;
    vector<int> vl = v, vr;
    p(vl, vr, 0);
    cout << ans;
}
```

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 $2 \leqslant a \leqslant 10^6$, $0 \leqslant b \leqslant 10^6$.

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++

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

```
int binpow (int a, int n) {
        int res = 1;
        while (n > 0) {
                if (n % 2 == 1)
                        (res *= a) %= MOD;
                (a *= a) %= MOD;
                n /= 2;
        }
        return res;
}

int bc(int n, int k){
    int res = f[n];
    int p1 = binpow(f[k], MOD - 2);
    int p2 = binpow(f[n - k], MOD - 2);
    (res *= p1) %= MOD;
    (res *= p2) %= MOD;
    return res;
}
signed main(){
    for(int i = 1; i <= 2e6; i++){
        f[i] = (f[i - 1] * i) % MOD;
    }
    int a, b;
    int ans = 0;
    cin >> a >> b;
    a--;
    for(int i = 0; i < a + 1; i++){
        if(2 * i <= b){
            int d = bc(a, i);
            if(b - 2 * i <= a - i){
                (d *= bc(a - i, b - 2 * i) ) %= MOD;
                (ans += d) %= MOD;
            }
        }
    }
    cout << ans << endl;
}
```

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 $1 \leqslant d \leqslant 200$, $2 \leqslant n \leqslant 2\,000$.

Во второй строке находятся n целых чисел v_i через пробел — скорости каждого джентльмена, где $1 \leqslant v_i \leqslant 2\,000$. Гарантируется, что все скорости попарно различны. Скорости даны в порядке возрастания, то есть $v_1 < v_2 < \ldots < v_n$.

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

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

Стандартный ввод
5 4 2 5 8 10 7
Стандартный вывод
0.556

Пример №3

Стандартный ввод
5 4 2 5 8 10 11
Стандартный вывод
1.000

Пример №4

Стандартный ввод
5 4 2 5 8 10 15
Стандартный вывод
1.429

Пример №5

Стандартный ввод
5 4 2 5 8 10 17
Стандартный вывод
1.667

Пример №6

Стандартный ввод
5 4 2 5 8 10 19
Стандартный вывод
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.

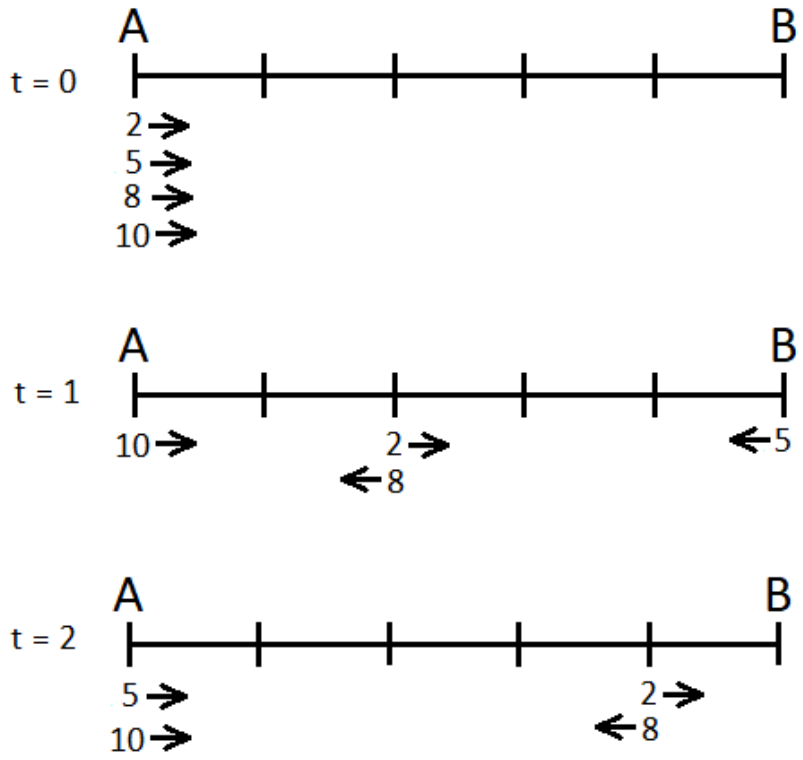


Рис. 2.2.8

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 встретятся 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++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const double EPS = 1e-7;
double x(double M, int V, int d){
    double dst = V * M;
    int cnt = floor((dst + EPS) / d);
    double pin = dst - cnt * d;
    if(cnt % 2 == 0){
        return pin;
    }
    else{
        return d - pin;
    }
}
int F(double M, vector<int> &v, int d){
    int res = 0;
    for(int i = 0; i < v.size(); i++){
        double dst = v[i] * M;
```

```
        int cnt = floor((dst + EPS) / d);
        res += cnt * i;
        double tx = x(M, v[i], d);
        for(int j = 0; j < i; j++){
            double txj = x(M, v[j], d);
            if(cnt % 2 == 0){
                res += txj <= tx + EPS;
            }
            else{
                res += txj >= tx - EPS;
            }
        }
    }
    return res;
}
signed main(){
    int d, n;
    cin >> d >> n;
    vector<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    int k;
    cin >> k;
    double L = 0, R = 1;
    while(F(R, v, d) <= k){
        R *= 2;
    }
    R *= 2;
    while(R - L > 1e-4){
        double M = (R + L) / 2.0;
        if(F(M, v, d) < k){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout.precision(10);
    cout << fixed << L << endl;
}
```

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.Контест: https://contest.yandex.ru/contest/63459/enter/.

Задача 2.3.1.1. (15 баллов)

Тема: арифметика.

Условие

Оля в каждую клетку таблицы 3×3 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 2×2 равна 32. Какое число Оля записала в центральную клетку таблицы?

Решение

Проанализируем исходную таблицу и увидим, что при построении всех возможных квадратов 2×2:

- 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 2×2 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 его 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

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

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 $\dfrac{192}{900\,000}\approx 0{,}0002$.

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

Двузначное 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 человек 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

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

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)

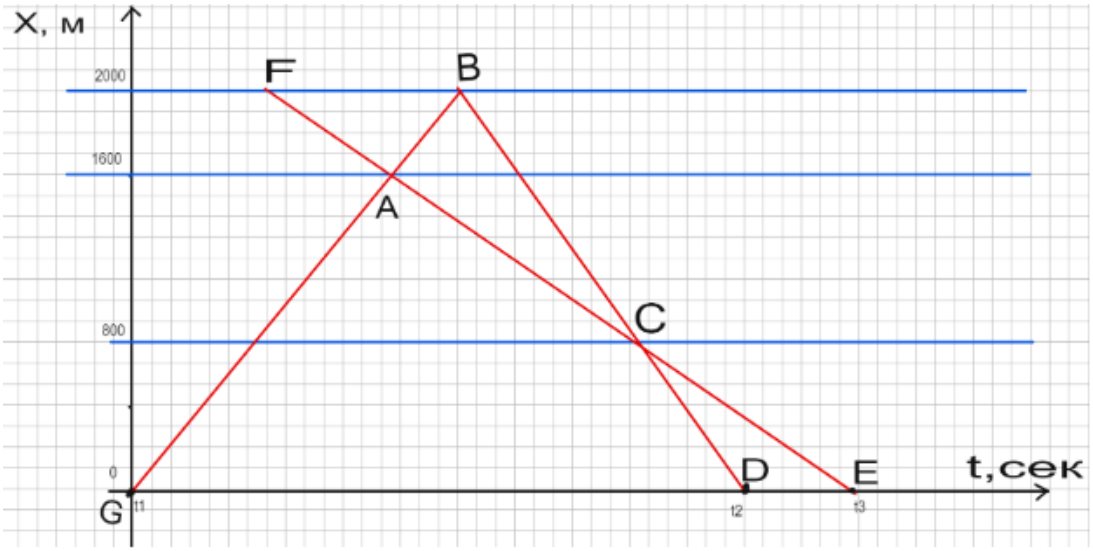
Темы: текстовые задачи и логика, графическое изображение движения, теорема Менелая.

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 скорость. 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:

$h_1 = 2\,000 - 400 = 1\,600$, $h(2) = 800$, $H = 2\,000$, $t_1 = 6$, $t_2 = 12$.

По теореме Менелая получаем:

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

Выразим эти отношения из разных пар подобных треугольников:

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

Подставим числа:

$$\frac{1\,600}{400} \cdot \frac{1\,200}{800} \cdot \frac{t_3 - 11}{t_3 - 6} = 1;$$
$$6 \cdot (t_3 - 11) = t_3 - 6;$$
$$5 \cdot t_3 = 60;$$
$$t_3 = 12 \text{ ч.}$$

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), позиции 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 $0 - 1 - 2 - 3 - 4 - 0 - \ldots$ (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, $\angle ABC = \angle ABD = \angle AED = 90°$.

Ответ: 90°.

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 $a + b = 11(x + y) \vdots 5$ и числа 5 и 11 взаимно просты, то

$$(x + y) \vdots 5. \tag{2.3.1}$$

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

$$(x - y) \vdots 3. \tag{2.3.2}$$

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, $y\vdots 11$ 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, 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 $Q = \dfrac{P}{2}$ 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 $Q = \dfrac{P}{2}$. 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.

Третьи 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

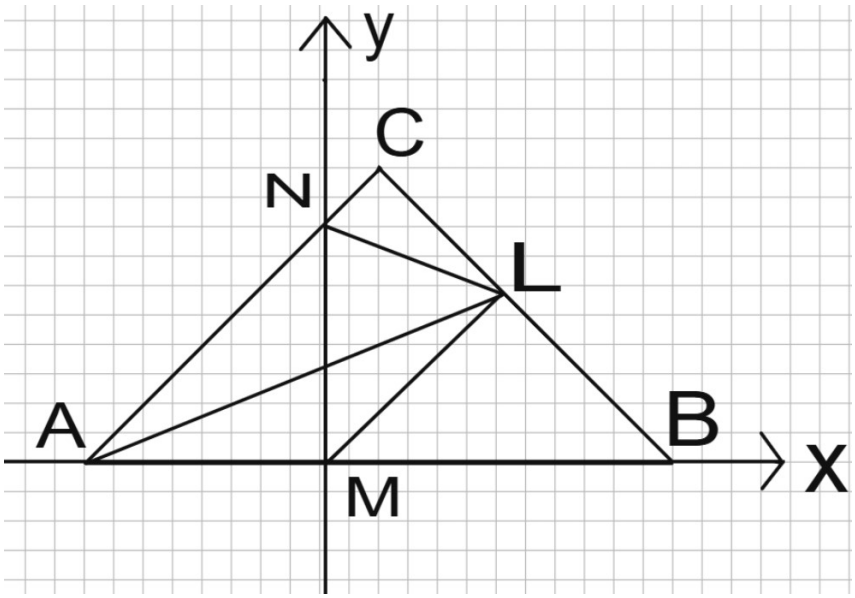
$MN = AM$, значит, угол ANM тоже равен 45° и NM перпендикулярно AB.

Тогда углы NML и BML тоже равны по 45°.

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:

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

Тогда угол ANL равен 180° 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:

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

$$16 \cdot x^2 - a \cdot x + x_0 \cdot a - 16 \cdot (x_0)^2 = 0;$$

$$D = a^2 - 4 \cdot 16 \cdot (x_0 \cdot a - 16 \cdot x_0^2) = (a - 32 \cdot x_0)^2 = 0;$$

$$a = 32 \cdot x_0.$$

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 $x_1 = \dfrac{x_0}{2}$.

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.Контест: 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

$$1{,}08x = 5\,832x = 5\,400.$$

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: комбинаторика.

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 разделю 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:

$$391 = xm + y,$$
$$604 = xk + y,$$
$$888 = xn + y,$$

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

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

$$284 = x(n - k),$$
$$213 = x(k - m).$$

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.

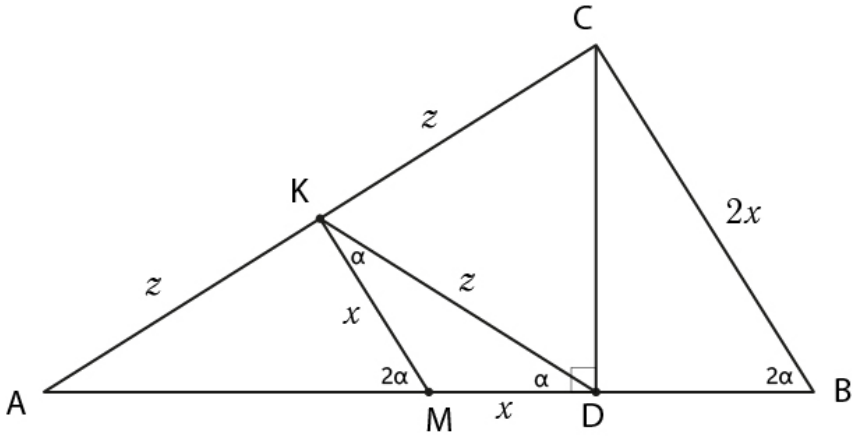
Задача 2.3.5.5. (25 баллов)

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: $DK = \dfrac{1}{2}AC = AK$.

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 Ф 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 Ф, равно дроби, в числителе которой шесть, а в знаменателе квадратный корень из числа π.

Чему равна площадь фигуры Ф?

Решение

1. Докажем, что все точки на границе фигуры равноудалены от центра симметрии O.
   - Возьмем на границе фигуры произвольную точку A. Пусть расстояние $OA = R$.
   - Возьмем любую другую точку B на границе фигуры.
   - Построим прямую I, проходящую через точку O и являющуюся биссектрисой угла AOB.
   - По свойству осевой симметрии, точка A', симметричная точке A относительно прямой I, также принадлежит фигуре Ф.
   - Поскольку I — биссектриса, точка A' попадет на луч OB. Так как при симметрии расстояние до центра сохраняется ($OA' = OA = R$), точка A' совпадет с точкой B только если $OB = R$.
   - Предположим, что $OB > R$. Тогда точка A лежит внутри отрезка OB. Но поскольку фигура выпуклая, весь отрезок OB должен принадлежать фигуре, а значит, и точка A не может быть граничной. Пришли к противоречию.
   - Предположим, что $OB < R$. Тогда точка B лежит внутри отрезка OA. Это также противоречит тому, что B — граничная точка.
   - Следовательно, единственно возможный вариант — $OB = R$.
   - Поскольку точка B была выбрана на границе произвольно, получается, что все точки границы фигуры Ф находятся на одинаковом расстоянии R от точки O. По определению, это окружность.
2. Так как границей является окружность, то сама фигура — круг.
3. Используя данное в условии значение диаметра ($6/\sqrt{\pi}$), находим радиус ($3/\sqrt{\pi}$) и вычисляем площадь, которая 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 $900 : 0{,}2 = 4\,500$ г анобтаниума.

Найдем массу топлива к началу первого часа.

Не было израсходовано главным двигателем к этому моменту $4\,500 + 300 = 4\,800$ г, что составляет $100 - 40 = 60\%$.

Составим пропорцию и решим ее:

$$\begin{aligned} 60\% &- 4\,800, \\ 100\% &- ? \end{aligned}$$

Значит, к началу первого часа было $4\,800 : 0{,}6 = 8\,000$ 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.

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

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:

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

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)

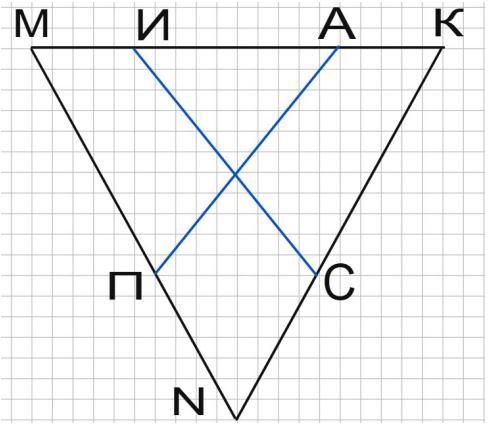
Темы: равносторонний треугольник, первый признак равенства треугольников.

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:

МА + МП = КИ + КС, а значит, МИ + МП = КА + КС = 1 000 м.

МИ + КА = ИА = 500 м. Кроме того, длина КМ равна 1 000 м.

Отсюда выходит, что КА = 1 000 − КС = 1000 − МА, а значит, КС = МА. Аналогично выходит, что КИ = МП.

Тогда треугольники КИС и МПА равны друг другу по двум сторонам и углу между ними.

Следовательно, АП = СИ, а значит, АП − СИ + 0,5 · ИА = 250 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 $7 \cdot 1\,440 = 10\,080 > 10\,000$.

Следовательно, под все условия подходит только число 1 440.

Ответ: 1 440.

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 задач

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca};$$
$$100a + 10b + c = 10a + b + 10b + c + 10c + a;$$
$$100a + 10b + c = 11a + 11b + 11c;$$
$$89a = 10c + b.$$

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

$$x^2 + y^2 = (x + y)^2 - 2xy = (2a + 1)^2 - 2(4a^2 + 8a - 4) = 4a^2 + 4a + 1 - 8a^2 - 16a + 8 =$$

$$= -4a^2 - 12a + 9 = -(4a^2 + 12a + 9) + 18 = -(2a + 3)^2 + 18.$$

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.

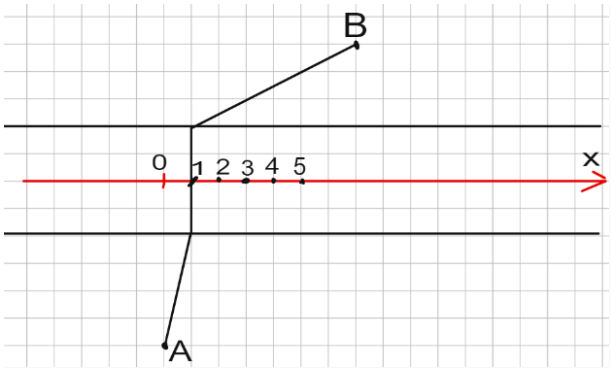


Рис. 2.3.1

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.

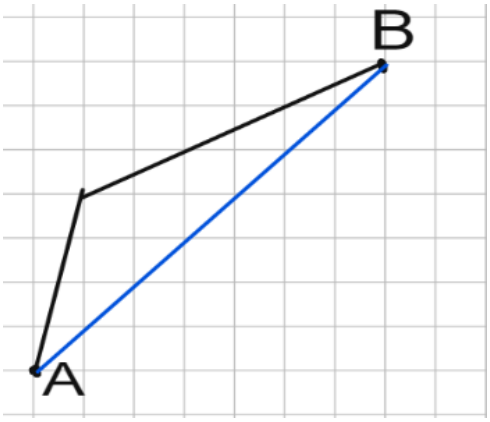


Рис. 2.3.2

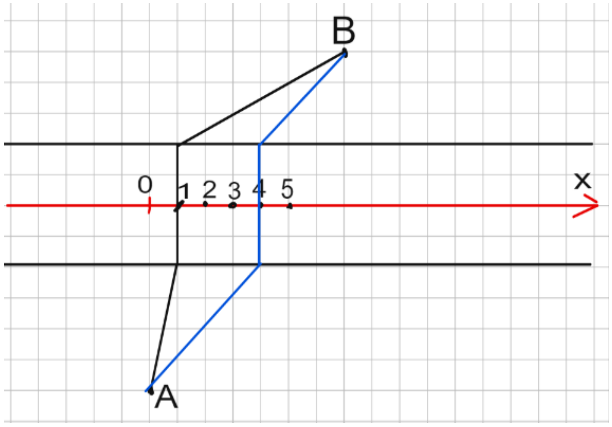


Рис. 2.3.3

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.

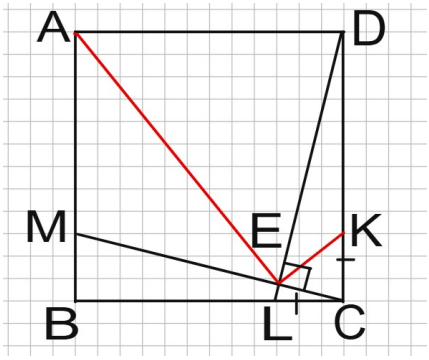


Рис. 2.3.4

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

Тогда

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

Так как вектора LE и LD должны быть сонаправлены, то

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

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

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

В профиле Разработка мобильных приложений через всю Олимпиаду проходит «Деловая Стажировка». Итак, представим, что участник Олимпиады хочет летом попасть на стажировку в компанию S-Mobile на должность Android-разработчика, а для этого ему предстоит пройти конкурсный отбор. Отбор будет состоять из двух этапов: индивидуальное собеседование и практическое задание.

Вначале предстоит пройти индивидуальное собеседование. У компании S-Mobile есть свой онлайн-магазин с названием S-Store. Кандидату на должность предстоит разобраться в его устройстве и ответить на вопросы. Начнем...

2.4.1. SQL

Любой онлайн-магазин обязательно хранит много информации: о товарах, продавцах, покупателях и многом другом. Такая информация чаще всего содержится в реляционной базе данных. В нашем случае — в базе данных `SQLite`. Дан фрагмент схемы базы данных магазина S-Store: https://drawsql.app/teams/myteam-1108/diagrams/e-store-database.

Задача: продемонстрировать навыки работы с реляционной базой данных с помощью `SQL` (https://en.wikipedia.org/wiki/SQL), а именно, базовые `CRUD`-операции (https://en.wikipedia.org/wiki/Create,_read,_update_and_delete).

Задача 2.4.1.1. Запрос на выборку данных. Перетаскивание (1 балл)

Тема: *SQL.*

Условие

Составьте `SQL`-запрос на выборку названия товара с первичным ключом, равным 5.

Элементы для заполнения пропусков: `SELECT, UPDATE, INSERT, FROM, WHERE, name, products, id, 5`.

_____ products._____

_____ _____

_____ _____.id = _____

Решение

Запрос на выборку осуществляется с помощью команды `SELECT`.

Указание таблиц, из которых происходит выборка данных, осуществляется с помощью команды FROM.

Указания условий, которым должны удовлетворять выбираемые записи, осуществляется с помощью ключевого слова WHERE.

Таким образом, запрос должен выглядеть так:

SQL

```sql
SELECT products.name
FROM products
WHERE products.id = 5
```

Задача 2.4.1.2. Запрос на выборку данных. Ввод текста (10 баллов)

Тема: SQL.

Условие

Определите количество товаров с ценой в диапазоне от A до B.

Решение

Запрос на выборку осуществляется с помощью команды SELECT.

Указание таблиц, из которых происходит выборка данных, осуществляется с помощью команды FROM.

Указания условий, которым должны удовлетворять выбираемые записи, осуществляется с помощью ключевого слова WHERE.

Для объединения условий используются логические операции AND, OR, NOT.

Таким образом, запрос может выглядеть так:

SQL

```sql
SELECT *
FROM products
WHERE products.price >= A AND products.price <= B
```

После этого можно посчитать количество строк, которые вернул запрос.

Также можно воспользоваться агрегирующей функцией COUNT. Тогда запрос будет выглядеть так:

SQL

```sql
SELECT COUNT(*)
FROM products
WHERE products.price >= A AND products.price <= B
```

Задача 2.4.1.3. Запрос на добавление данных. Ввод и выполнение запроса (10 баллов)

Тема: SQL.

Условие

Составьте SQL-запрос, который добавит в базу данных новый продукт со следующими характеристиками.

- Название продукта: `NAME` (задается случайно).
- Описание продукта: `DESC` (задается случайно).
- Цена: `PRICE` (задается случайно).
- Категория товара: `CATEGORY_NAME` (задается случайно).

Никаких других записей НЕ нужно добавлять, модифицировать или удалять.

Решение

Вначале необходимо узнать первичный ключ категории товара, для этого можно воспользоваться запросом:

SQL

```sql
SELECT *
FROM CATEGORY_NAME
WHERE name = CATEGORY_NAME
```

Здесь уточнение условий, которым должны удовлетворять выбираемые записи, осуществляется с помощью ключевого слова `WHERE`.

Запрос на добавление данных осуществляется с помощью команды `INSERT INTO`, за которой указывается название таблицы и полей.

Значения полей указывается после команды `VALUES`.

SQL

```sql
INSERT INTO product (name, description, price, category_id)
VALUES(NAME, DESC, PRICE, category_id);
```

Ответ. Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной верно, если после этого в базе данных присутствует запись, которую необходимо было добавить в соответствии с условием задачи.

Задача 2.4.1.4. Запрос на обновление данных. Ввод и выполнение запроса (10 баллов)

Тема: SQL.

Условие

Составьте SQL-запрос на изменение в базе данных цены продукта, который создали в прошлом задании. Новая цена продукта должна стать NEW_PRICE.

Никаких других записей не нужно добавлять, модифицировать или удалять.

Решение

Запрос на обновление данных осуществляется с помощью команды UPDATE.

Поля и их новые значения указываются после команды SET.

Указания условий, которым должны удовлетворять обновляемые записи, осуществляется с помощью ключевого слова WHERE.

SQL

```
1 UPDATE products
2 SET price = NEW_PRICE
3 WHERE id = ID
```

Ответ. Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной верно, если после этого в базе данных присутствует запись, которую необходимо было изменить в соответствии с условием задачи.

Задача 2.4.1.5. Запрос на удаление данных. Ввод и выполнение запроса (10 баллов)

Тема: SQL.

Условие

Составьте SQL-запрос на удаление из базы данных продукта, который создали ранее в задаче 2.4.1.3.

Решение

Запрос на удаление данных осуществляется с помощью команды DELETE FROM.

Указания условий, которым должны удовлетворять обновляемые записи, осуществляется с помощью ключевого слова WHERE.

SQL

```
1 DELETE FROM product
2 WHERE id = ID
```

Ответ. Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной

верно, если после этого в базе данных присутствует запись, которую необходимо было удалить в соответствии с условием задачи.

2.4.2. JSON

Бэкенд онлайн-магазина S-Store предоставляет данные клиентскому `Android`-приложению в формате `JSON` (https://en.wikipedia.org/wiki/JSON). Поэтому следующий блок вопросов нашего собеседования будет посвящен именно этому стандарту обмена данными.

Задача 2.4.2.1. JSON продукта. Перетаскивание (1 балл)

Темы: JSON, SQL.

Условие

Представьте информацию о товаре с `ID`, равным 70, из базы данных в формате `JSON`.

Элементы для перемещения: `1,2,name,inventory_id,799.99,899.99,2024-08-09T12:00:00,created_at,2025-08-09T12:00:00`.

```
{
  "___": 70,
  "___": "Smartphone XYZ",
  "description": "Latest model with advanced features.",
  "SKU": "ELEC-001",
  "category_id": 1,
  "___": 1,
  "price": ___,
  "discount_id": null,
  "___": "___",
  "modified_at": "2024-08-09T12:00:00"
}
```

Решение

Получаем информацию аналогично решению в задании 2.4.1 и составляем `JSON` этого объекта.

Ответ.

```
{
    "id": 70,
    "name": "Smartphone XYZ",
    "description": "Latest model with advanced features.",
    "SKU": "ELEC-001",
    "category_id": 1,
    "inventory_id": 1,
    "price": 799.99,
```

```
    "discount_id": null,
    "created_at": "2024-08-09T12:00:00",
    "modified_at": "2024-08-09T12:00:00"
}
```

Задача 2.4.2.2. JSON продукта. Ввод JSON (5 баллов)

Темы: JSON, Kotlin.

Условие

Android-приложение S-Store написано на языке программирования Kotlin с использованием Android SDK. Ниже представлен data-класс, представляющий ИМЯ_СУЩНОСТИ (рандомно). Составьте JSON-описание, соответствующее этому классу. Значения свойств задать произвольно.

Пример одного из вариантов.

Kotlin

```kotlin
data class User(
    val id: Int,
    val username: String?,
    val password: String?,
    val firstName: String?,
    val lastName: String?,
    val telephone: Int?,
    val createdAt: LocalDateTime?,
    val modifiedAt: LocalDateTime?,
    val createdBy: Int
)
```

Решение

В соответствии с кодом класса выбираем необходимые типы данных и составляем JSON.

Ответ.

```
{
  "id": 1,
  "username": "johndoe",
  "password": "securepassword123",
  "firstName": "John",
  "lastName": "Doe",
  "telephone": 1234567890,
  "createdAt": "2024-08-09T12:00:00",
  "modifiedAt": "2024-08-09T12:30:00",
  "createdBy": 100
}
```

Задача 2.4.2.3. Класс из JSON. Ввод кода класса (5 баллов)

Темы: JSON, Java.

Условие

Бэкенд магазина разработан на языке Java с использованием фреймворка `Spring Boot`. Вот JSON, представляющий ИМЯ_СУЩНОСТИ (задается случайно). Составьте класс на языке программирования Java, соответствующий этому `JSON`. Все поля должны быть с модификатором доступа `private`. Каждому полю должен соответствовать `getter` и `setter` с модификатором доступа `public`.

Далее пример одного из вариантов.

```
{
  "id": 1,
  "username": "johndoe",
  "password": "securepassword123",
  "firstName": "John",
  "lastName": "Doe",
  "telephone": 1234567890,
  "createdAt": "2024-08-09T12:00:00",
  "modifiedAt": "2024-08-09T12:30:00",
  "createdBy": 100
}
```

Решение

В соответствии с кодом `JSON` выбираем необходимые типы данных и составляем код класса на языке Java.

Ниже представлено решение на языке Java.

Java

```
public class User {

    private int id;
    private String username;
    private String password;
    private String firstName;
    private String lastName;
    private Integer telephone;
    private LocalDateTime createdAt;
    private LocalDateTime modifiedAt;
    private int createdBy;

    // Default constructor
    public User() {
    }

    // Constructor with all fields
    public User(int id, String username, String password, String
    ↪  firstName, String lastName, Integer telephone, LocalDateTime
    ↪  createdAt, LocalDateTime modifiedAt, int createdBy) {
        this.id = id;
        this.username = username;
```

```
21          this.password = password;
22          this.firstName = firstName;
23          this.lastName = lastName;
24          this.telephone = telephone;
25          this.createdAt = createdAt;
26          this.modifiedAt = modifiedAt;
27          this.createdBy = createdBy;
28      }
29
30      // Getters and Setters
31      public int getId() {
32          return id;
33      }
34
35      public void setId(int id) {
36          this.id = id;
37      }
38
39      public String getUsername() {
40          return username;
41      }
42
43      public void setUsername(String username) {
44          this.username = username;
45      }
46
47      public String getPassword() {
48          return password;
49      }
50
51      public void setPassword(String password) {
52          this.password = password;
53      }
54
55      public String getFirstName() {
56          return firstName;
57      }
58
59      public void setFirstName(String firstName) {
60          this.firstName = firstName;
61      }
62
63      public String getLastName() {
64          return lastName;
65      }
66
67      public void setLastName(String lastName) {
68          this.lastName = lastName;
69      }
70
71      public Integer getTelephone() {
72          return telephone;
73      }
74
75      public void setTelephone(Integer telephone) {
76          this.telephone = telephone;
77      }
78
79      public LocalDateTime getCreatedAt() {
80          return createdAt;
```

```
81      }
82
83      public void setCreatedAt(LocalDateTime createdAt) {
84          this.createdAt = createdAt;
85      }
86
87      public LocalDateTime getModifiedAt() {
88          return modifiedAt;
89      }
90
91      public void setModifiedAt(LocalDateTime modifiedAt) {
92          this.modifiedAt = modifiedAt;
93      }
94
95      public int getCreatedBy() {
96          return createdBy;
97      }
98
99      public void setCreatedBy(int createdBy) {
100         this.createdBy = createdBy;
101     }
102
103     // toString method
104     @Override
105     public String toString() {
106         return "User{" +
107                 "id=" + id +
108                 ", username='" + username + '\'' +
109                 ", password='" + password + '\'' +
110                 ", firstName='" + firstName + '\'' +
111                 ", lastName='" + lastName + '\'' +
112                 ", telephone=" + telephone +
113                 ", createdAt=" + createdAt +
114                 ", modifiedAt=" + modifiedAt +
115                 ", createdBy=" + createdBy +
116                 '}';
117     }
118 }
```

2.4.3. Запросы на сервер

Бэкенд магазина S-Store предоставляет клиентскому Android-приложению API. По ссылке можно найти Swagger (https://en.wikipedia.org/wiki/Swagger_(software)) документацию API для работы с пользователями. Изучив документацию, составьте HTTP-запросы, которые реализуют CRUD-операции. Эти запросы можно составлять и отправлять, например, с помощью Postman: https://www.postman.com.

Задача 2.4.3.1. Запрос получение данных. Ввод текста (1 балл)

Тема: API.

Условие

Составить HTTP-запрос на выборку пользователя с первичным ключом, равным ID (задается случайно). В качестве ответа укажите фамилию пользователя.

Решение

Анализируем документацию API, находим соответствующий запрос и составляем его.

Ответ.

Зависит от случайно сгенерированного параметра ID.

Общий ответ для всех вариантов:

ФАМИЛИЯ

Например, если ID=1, то ответ будет:

Doe

Задача 2.4.3.2. Запрос на добавление данных. Ввод и выполнение запроса (10 баллов)

Тема: API.

Условие

Добавьте в базу данных нового пользователя со следующими характеристиками:

- Название продукта: name.
- Описание продукта: desc.
- Цена: price.
- Категория товара: category_name.

Решение

Анализируем документацию API, находим соответствующий запрос, составляем его и отправляем с помощью Postman.

Критерии оценивания

Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной верно, если после этого в базе данных присутствует запись, которую необходимо было добавить в соответствии с условием задачи.

Задача 2.4.3.3. Запрос на обновление данных. Ввод и выполнение запроса (10 баллов)

Тема: API.

Условие

Измените в базе данных контактный телефон пользователя, который создали в задаче 2.4.3.2. Новый телефон должен стать `new_telephone` (рандомно).

Решение

Анализируем документацию `API`, находим соответствующий запрос, составляем его и отправляем с помощью `Postman`.

Критерии оценивания

Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной верно, если после этого в базе данных присутствует запись, которую необходимо было изменить в соответствии с условием задачи.

Задача 2.4.3.4. Запрос на удаление данных. Ввод и выполнение запроса (10 баллов)

Тема: API.

Условие

Удалите из базы данных пользователя, которого создали ранее в задаче 2.4.3.2.

Решение

Анализируем документацию `API`, находим соответствующий запрос, составляем его и отправляем с помощью `Postman`.

Критерии оценивания

Правильность выполнения определяется конечным состоянием базы данных после выполнения запроса, составленного участником. Задача считается решенной верно, если после этого в базе данных присутствует запись, которую необходимо было добавить в соответствии с условием задачи.

2.4.4. IDE, гит, анализ ошибки

Задача 2.4.4.1. Ввод github аккаунта. Ввод текста (1 балл)

Тема: git.

Условие

Составить `HTTP`-запрос на выборку пользователя с первичным ключом, равным `id` (задается случайно). В качестве ответа укажите фамилию пользователя.

Критерии оценивания

Проверяется автоматически, что в профиле пользователя заполнено нужное поле и оно содержит ссылку на `github`-аккаунт.

Задача 2.4.4.2. Клонирование репозитория (0 баллов)

Тема: git.

Методика проверки

Баллы не зачисляются.

Условие

Склонируйте ветку `BRANCH_NAME` (задается случайно) репозитория, в котором находится код мобильного приложения S-Store в свой аккаунт и откройте код в `Android Studio`.

Критерии оценивания

Данное задание не проверяется и не оценивается. Это руководство к действию.

Задача 2.4.4.3. Анализ ошибки. Выбор пропущенных слов (5 баллов)

Тема: анализ кода.

Условие

Попробуем запустить приложение. Очевидно, что программа завершилась ошибкой. Так и было задумано!

При программировании под Android ошибки можно увидеть в специальном журнале (LogCat). По нему программист определяет, какая произошла ошибка и в какой строке.

Укажите файл, строку, в которой произошла ошибка и вид ошибки.

Ошибка содержалась в файле (MainActivity.kt, AndroidManifest.xml, buil.gradle.kts) в строке (одно число).

Причиной этого стало (арифметическая операция, переполнение стека, нехватка памяти, зависание программы, обращение к несуществующему объекту).

Решение

В зависимости от BRANCH_NAME, которую надо было клонировать, возможны разные ошибки.

Ответ: ошибка содержалась в файле MainActivity.kt в строке 34. Причиной этого стало (арифметическая операция, переполнение стека, нехватка памяти, зависание программы, обращение к несуществующему объекту).

2.4.5. XML

Android SDK активно использует XML для хранения различных данных приложения, например, разметку интерфейса, темы интерфейса, систему навигации, иконки, строковые ресурсы, информацию, которую приложение предоставляет операционной системе Android (манифест приложения) и многое другое. В этом блоке нужно найти в xml-файлах приложения S-Store необходимую информацию о приложении.

Задача 2.4.5.1. Разрешения приложения. На соответствие (1 балл)

Тема: xml.

Условие

Определите, какие разрешения может запросить у операционной системы приложение S-Store. В качестве ответа сопоставьте название разрешения с его описанием. Если разрешение отсутствует в манифесте, укажите это.

Варианты зависят от BRANCH_NAME, которую нужно было клонировать. Здесь представлен вариант для ветки BRANCH_NAME=var1.

Таблица 2.4.1

Разрешения	**Варианты описания разрешений (порядок в учебной системе будет рандомный)**

BLUETOOTH	Позволяет приложениям подключаться к сопряженным устройствам Bluetooth
ACCESS_NETWORK_STATE	Позволяет приложениям получать доступ к информации о сетях
ACTIVITY_RECOGNITION	Позволяет приложению распознавать физическую активность
INTERNET	Позволяет приложениям открывать сетевые сокеты
READ_CONTACTS	Позволяет приложению считывать контактные данные пользователя
	Разрешение отсутствует в манифесте

Решение

Надо открыть файл манифеста и определить, какие разрешение запрашивает приложение. В разных BRANCH_NAME будут разные разрешения.

Ответ.

Таблица 2.4.2

Разрешения	**Описание разрешения**
BLUETOOTH	Позволяет приложениям подключаться к сопряженным устройствам Bluetooth
ACCESS_NETWORK_STATE	Позволяет приложениям получать доступ к информации о сетях
ACTIVITY_RECOGNITION	Позволяет приложению распознавать физическую активность
INTERNET	Разрешение отсутствует в манифесте
READ_CONTACTS	Разрешение отсутствует в манифесте

Задача 2.4.5.2. Цвет. Ввод текста (5 баллов)

Темы: XML, представление цветов в формате RGB в шестнадцатеричной и десятичной системах счисления.

Условие

Определите цвет основного ELEMENT (задается случайно) приложения в THEM E_COLOR теме (задается случайно). Ответ представьте в виде четырех чисел, введенных через пробел, представляющих компоненты Red, Green, Blue и Alpha соответственно. Каждое число должно быть в десятичной системе счисления, в диапазоне от 0 до 255.

Решение

Надо открыть XML-файлы темы и цветов, сопоставить их и определить цвет, после чего перевести его в десятичную систему счисления.

Ответ.

Зависит от случайно сгенерированной пары (ELEMENT, THEME_COLOR).

Общий ответ для всех вариантов:

R G B A

Например, если (ELEMENT, THEME_COLOR)=(Action bar, Dark), то ответ для BRANCH_NAME=var1 будет:

47 91 237 0

Задача 2.4.5.3. Интернационализация. Ввод XML (5 баллов)

Темы: XML, представление цветов в формате RGB в шестнадцатеричной и десятичной системах счисления.

Условие

Интерфейс Android приложение магазина S-Store может отображаться на двух языках: русском и английском. Однако перевод приложения на английский язык еще не завершен. Составьте фрагмент XML, который бы позволил отображаться на английском языке строковый ресурс RESOURCE_NAME (задается случайно).

Решение

Надо открыть XML строковых ресурсов и перевести на английский язык соответствующий элемент, после чего составить необходимый XML-элемент для этого же ресурса на английском языке.

Ответ.

Зависит от случайно сгенерированного параметра RESOURCE_NAME.

Общий ответ для всех вариантов:

```
<string name="RESOURCE_NAME"> RESOURCE_NAME_TRANSLATION</string>
```

Например, если RESOURCE_NAME=security, то ответ будет:

```
<string name="security">Security</string>
```

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.
- **Командная работа.** Одной из ключевых задач наставника на втором этапе является помощь 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

Задачи второго этапа максимально приближены к реальным задачам разработки мобильных приложений, предоставляя участникам возможность испытать свои навыки в этой сфере.

3.2.1. Индивидуальные задачи

Перед выполнением индивидуальных заданий участник должен выбрать роль задания, которую он будет выполнять. Возможные роли: Android-разработчик, UI/UX-дизайнер, backend-разработчик. Выбрать можно только одну из ролей. Для каждой роли будут даны три задачи легкого, среднего и высокого уровня сложности.

Каждое задание для ролей Android-разработчика и UI/UX-дизайнера представляют собой заготовку Android-приложения, которую нужно доработать в соответствие с техническим заданием. Заготовка приложения размещена в `git`-репозитории, а техническое задание содержится в `read.me` этого репозитория, поэтому текст задания в учебной системе содержит только ссылку на этот репозиторий.

В качестве решения участник сдает в учебную систему zip-архив со всеми файлами проекта.

Решение участника проверяется автоматически с помощью UI-тестов.

Задача 3.2.1.1. Подарочек (2 балла)

Тема: UI мобильного приложения.

Условие

Файлы: https://git.sicampus.ru/Java/present.

В задании предлагается дописать существующий проект Android-приложения, а именно, дополнить разметку.

- В разметке приложения в любой ориентации должны присутствовать элементы, указанные в таблице 3.2.2. У кнопки устанавливается строковый ресурс из таблицы 3.2.2.
- В портретной ориентации: в ImageView должен быть установлен ресурс `@drawable/orange`, кнопку следует расположить строго под картинкой.
- В ландшафтной ориентации: вместо оранжевой коробочки в ImageView должна отображаться коробочка синего цвета `@drawable/blue`, картинка — слева от кнопки.

Таблица 3.2.1. Элементы пользовательского интерфейса

№	View type	id
0	Button	test_button
1	ImageView	box_image

Таблица 3.2.2. `strings.xml`

№	Ресурс	Допустимое значение
0	`@string/main_text_button`	Test Button

Решение

https://git.sicampus.ru/Java/present-solve.

Задача 3.2.1.2. Радуга (3 балла)

Тема: UI мобильного приложения.

Условие

Файлы: https://git.sicampus.ru/Java/rainbow.

Дополните разметку следующими элементами:

- Элемент `TextView` (`@id/main_text`), который содержит в себе текст с поддержкой языка по умолчанию и английского. Значения приведены в таблице 3.2.4. Элемент должен быть расположен всегда сверху экрана.
- Элемент-наследник от `ViewGroup` (например, `FrameLayout`, `LinearLayout` и т. п.) с идентификатором `@id/outer_layout`, включающим только элементы `View` в виде полос (вертикальные для альбомной ориентации и горизонтальные для портретной ориентации).
- Каждая полоса соответствует необходимому цвету радуги, разделенному между собой белым цветом (белый, красный, белый, оранжевый, белый, желтый, белый, зеленый, белый, голубой, белый, синий, белый, фиолетовый, белый).
- У каждой полосы с цветом должен быть идентификатор. Набор идентификаторов для цветов следующий:
  - `@id/red`;
  - `@id/orange`;
  - `@id/yellow`;
  - `@id/green`;
  - `@id/azure`;
  - `@id/blue`;
  - `@id/violet`;
  - `@id/white`.

Таблица 3.2.3. `colors.xml`

Ресурс	Значение
`@id/red`	#FF0000
`@id/orange`	#F6A630
`@id/yellow`	#FFEB3B
`@id/green`	#00FF00
`@id/azure`	#2196F3
`@id/blue`	#0000FF
`@id/violet`	#673AB7
`@id/white`	#FFFFFF

Таблица 3.2.4. `strings.xml`

Ресурс	Значение	Квалификатор
`@string/main_text`	каждый охотник желает знать где сидит фазан	
`@string/main_text`	Richard Of York Gave Battle In Vain	en

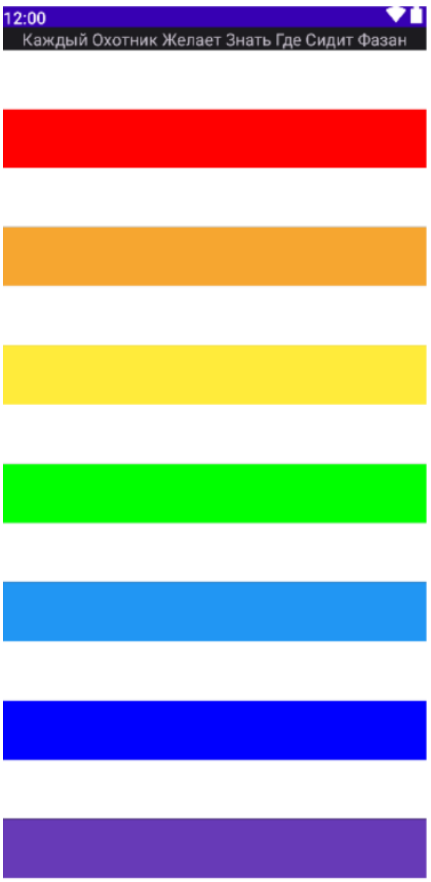


Рис. 3.2.1. Портретная ориентация

Рис. 3.2.2. Альбомная ориентация

Критерии оценивания

Таблица 3.2.5. Критерии оценивания и тесты

#	Тест	Балл	Проверка
1	`interfaceTest`	1	Accessibility Checks
2	`languageTest`	1	Строковые ресурсы на русском и английском языках
3	`checkValues`	1	Значения цветов
4	`checkPortrait`	2	Порядок следования цветов
5	`checkLandscape`	3	Порядок следования цветов

Решение

https://git.sicampus.ru/Java/rainbow-solve.

Задача 3.2.1.3. Ежедневник (5 баллов)

Тема: UI мобильного приложения.

Условие

Файлы: https://git.sicampus.ru/Java/diary.

В задании предлагается дописать существующий проект-заготовку Android-приложения, имитирующего ведение ежедневника.

Измените код существующих классов и интерфейсов, дополните разметку, ресурсы и манифест, чтобы приложение отвечало следующим требованиям:

- Интерфейс приложения должен содержать элементы, перечисленные в таблице 3.2.6.
- При запуске приложения все текстовые поля для ввода данных остаются пустыми, в них будет отображаться подсказка.

- По нажатии на кнопку **Сохранить** в случае отсутствия информации о названии события появляется **Snackbar** с текстом «Введите название события!» (без кавычек).
- По нажатии на кнопку **Сохранить** в случае указания названия события появляется диалоговое окно с кнопкой **ОК** и текстом, представленным ниже.

```
Записано!
Событие: title
Дата: date
Время: time
Заметки: notes
```

- Дата выводится в формате: дд.мм.гггг.
- После нажатия на кнопку **Сохранить** все текстовые поля очищаются.

Таблица 3.2.6. Элементы пользовательского интерфейса

#	View type	id	hint	text
0	`EditText`	`event_title_user_input`	Название события	
1	`EditText`	`event_time_user_input`	Время события	
2	`EditText`	`event_notes_user_input`	Заметки к событию	
3	`CalendarView`	произвольно	—	
4	`Button`	произвольно	—	Сохранить

Критерии оценивания

Таблица 3.2.7. Критерии оценивания и тесты

#	Тест	Балл	Проверка
1	`mainTest`	1	Показ `AlertDialog`
2	`emptyTest`	1	Очищение полей ввода
3	`hasHint`	1	Проверка наличия подсказок

Решение

https://git.sicampus.ru/Java/diary-solve.

Задача 3.2.1.4. Жизненный цикл Activity (2 балла)

Тема: Android-разработка.

Условие

Файлы: https://git.sicampus.ru/Java/activity-lifecycle.

Необходимо дополнить данное приложение отслеживанием состояний активности.

Основные требования к приложению:

- Не изменяйте место и название файла `MainActivity` (но редактирование содержимого (в том числе и уже имеющихся в файле методов) — можно и нужно).
- Последовательность состояния жизненого цикла приложения должна выводиться в `TextView` (`@id/protocol`).
- Каждое состояние жизненного цикла выводится в отдельной строке.
- Значения текстовых подсказок должны быть определены в строковых ресурсах `string.xml`.
- Текстовые подсказки должны быть корректно выводится при смене конфигурации приложения (например, поворот экрана).

Название строковых ресурсов и их значения приведено в таблице 3.2.8.

Таблица 3.2.8. Настройки всплывающих подсказок

№	Отслеживаемый метод	Строковый ресурс	Значение ресурса
1	onCreate	ncreate	Activity CREATED
2	onStart	nstart	Activity STARTED
3	onResume	nresume	Activity RESUMED
4	onPause	npause	Activity PAUSED
5	onStop	nstop	Activity STOPPED
6	onRestart	nrestart	Activity RESTARTED
7	onDestroy	ndestroy	Activity DESTROYED

Примеры

```
Запуск приложения
Входные данные: первый запуск приложения (после установки)
Ожидаемый результат:
TextView (@id/protocol) содержит три строки:
Activity CREATED
Activity STARTED
Activity RESUMED
```

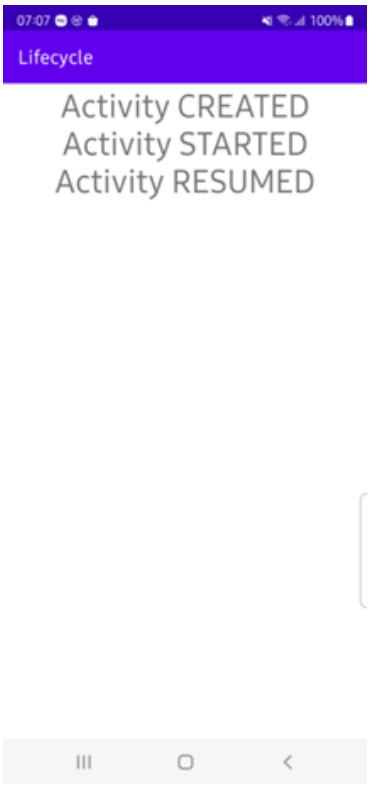


Рис. 3.2.3

Критерии оценивания

Таблица 3.2.9. Критерии оценивания и тесты

№	Название теста	Баллы	Описание
1	`checkPortait`	2	Проверка последовательности `Toast` при запуске активности
2	`checkLandscape`	4	Проверка обработки поворотов экрана

Решение

Решение: https://git.sicampus.ru/Java/activity-lifecycle-solve.

Задача 3.2.1.5. BroadcastReceiver (3 балла)

Тема: Android-разработка.

Условие

Файлы: https://git.sicampus.ru/Java/Broadcast-Receiver.

В данном практическом задании предлагается дополнить существующий проект-заготовку Android-приложения.

Советуем сначала склонировать шаблон проекта и открыть его в Android Studio, так как этот процесс может занимать продолжительное время.

Допишите код проекта, чтобы приложение отвечало следующим требованиям:

- Нестатический метод `makeBroadcastReceiver()`, определенный в классе `MainActivity`, возвращает объект `BroadcastReceiver`.
- Объект `BroadcastReceiver`, возвращаемый методом `makeBroadcastReceiver()`, способен обрабатывать широковещательные намерения `android.intent.action.*` (полный список констант представлен ниже).
- При запуске приложения в `TextView` с `id/status_text` может быть установлен произвольный текст. При получении широковещательного сообщения в `TextView` устанавливается текст, соответствующий типу `action` (как в списке ниже). Например, при получении намерения `Intent.ACTION_ASSIST` в `TextView` должен быть установлен текст `ACTION_ASSIST`.

Таблица 3.2.10. Элементы пользовательского интерфейса

View type	**id**
`TextView`	`status_text`

Константы `Intent`:

1. `ACTION_AIRPLANE_MODE_CHANGED`;
2. `ACTION_APPLICATION_RESTRICTIONS_CHANGED`;
3. `ACTION_APPLICATION_LOCALE_CHANGED`;
4. `ACTION_ASSIST`;
5. `ACTION_BATTERY_CHANGED`;
6. `ACTION_BATTERY_LOW`;
7. `ACTION_BATTERY_OKAY`;
8. `ACTION_CALL`;
9. `ACTION_DATE_CHANGED`;
10. `ACTION_DEFAULT`;
11. `ACTION_HEADSET_PLUG`;
12. `ACTION_LOCALE_CHANGED`;
13. `ACTION_TIME_TICK`.

Сдать в систему тестирования необходимо zip-архив, в корневом каталоге которого располагается дополненный проект Android Studio. После загрузки zip-архива приложение будет запущено в системе автоматического тестирования для проверки на соответствие техническому заданию.

При сборке проекта все изменения в файле `build.gradle` игнорируются — будут использованы зависимости только из шаблона-заготовки; не изменяйте пакет приложения.

Критерии оценивания

Таблица 3.2.11. Критерии оценивания и тесты

№	Тест	Балл	Проверка
1–13	`Intent.ACTION_*`	13	Обработка широковещательных намерений

Решение

Решение: https://git.sicampus.ru/Java/Broadcast-Receiver-solve.

Задача 3.2.1.6. Tricky Hexahedron (5 баллов)

Тема: Android-разработка.

Условие

Файлы: https://git.sicampus.ru/Java/tricky-hexahedron.

В данном практическом задании предлагается дописать существующий проект-заготовку Android-приложения для вычисления суммы длин сторон, длины диагонали, площади полной поверхности и объема прямоугольного параллелепипеда.

Измените код существующих классов и интерфейсов, дополните разметку, ресурсы и манифест, чтобы приложение отвечало следующим требованиям:

- Интерфейс приложения должен содержать элементы, перечисленные в таблице 3.2.12, чтобы они были видны полностью даже при открытой виртуальной клавиатуре.
- Элементы `EditText` обязаны поддерживать ввод чисел с плавающей точкой. Элемент `EditText` с большим номером по порядку располагается ниже `EditText` с меньшим номером.
- Ограничения на вводимые значения указаны в таблице 3.2.12. Точность проверки: 1e-5.
- Элемент `@id/spinner` позволяет выбрать одну из четырех опций для проведения расчетов в порядке, определенном в данном задании.
- При нажатии на элемент `@id/calculate` в текстовое поле `@id/solution` устанавливается текст с результатом проведенных расчетов в соответствии с выбранной в `@id/spinner` опцией. Гарантируется, что в момент нажатия все три `EditText` имеют непустой текст.
- При нажатии на `TextView` с `@id/solution` текущее значение копируется в буфер обмена.

Таблица 3.2.12. Элементы пользовательского интерфейса

View type	**id**	**Максимальное значение**
`TextView`	`side_a_label`	
`EditText`	`side_a`	10^{12}

View type	id	Максимальное значение
TextView	side_b_label	
EditText	side_b	10^{12}
TextView	side_c_label	
EditText	side_c	10^{12}
Spinner	spinner	
Button	calculate	
TextView	solution	

Таблица 3.2.13. strings.xml

Ресурс	Допустимое значение
@string/side_a_text	a
@string/side_b_text	b
@string/side_c_text	c

Решение

Решение: https://git.sicampus.ru/vladimir-shperling/ListOfWork/src/branch/main/details/Broadcast-Receiver-solve.zip.

Задача 3.2.1.7. Endpoint (2 балла)

Тема: backend-разработка.

Условие

Ниже представлен код одного из методов REST-контроллера фреймворка Spring Boot. Заполните пропуски в коде так, чтобы метод обрабатывал запрос соответствующий OpenAPI: https://innovationcampus.ru/store/swagger-ui.html.

Доступные для перемещения фрагменты: 200, 300, 404, 402, id, Get, Post, Add, Delete, User.

Java

```
@Operation(summary = "Get user with id")
@ApiResponses(value = {
        @ApiResponse(responseCode = "    ",
                description = "Get user with id",
                content = { @Content(mediaType = "application/json",
                        schema = @Schema(implementation = User.class))
                        ↪  }),
        @ApiResponse(responseCode = "   ",
                description = "No user found",
                content = @Content) })
```

```
@GetMapping("/user")
public Optional<    > getUser(@RequestParam(value = "  ", defaultValue
    = "1") Integer   ) throws UserNotFound {
    if (repo.existsById(id)) {
        return repo.findById(id);
    }
    else {
        User emptyUser = new User();
        emptyUser.setId(  );
        throw new UserNotFound("User with id %s not found", emptyUser);
    }
```

Задача 3.2.1.8. Entity класс (3 балла)

Тема: backend-разработка.

Условие

Реализуйте Entity-класс фреймворка Spring Boot соответствующий сущности user базы данных: https://drawsql.app/teams/myteam-1108/diagrams/e-store-database.

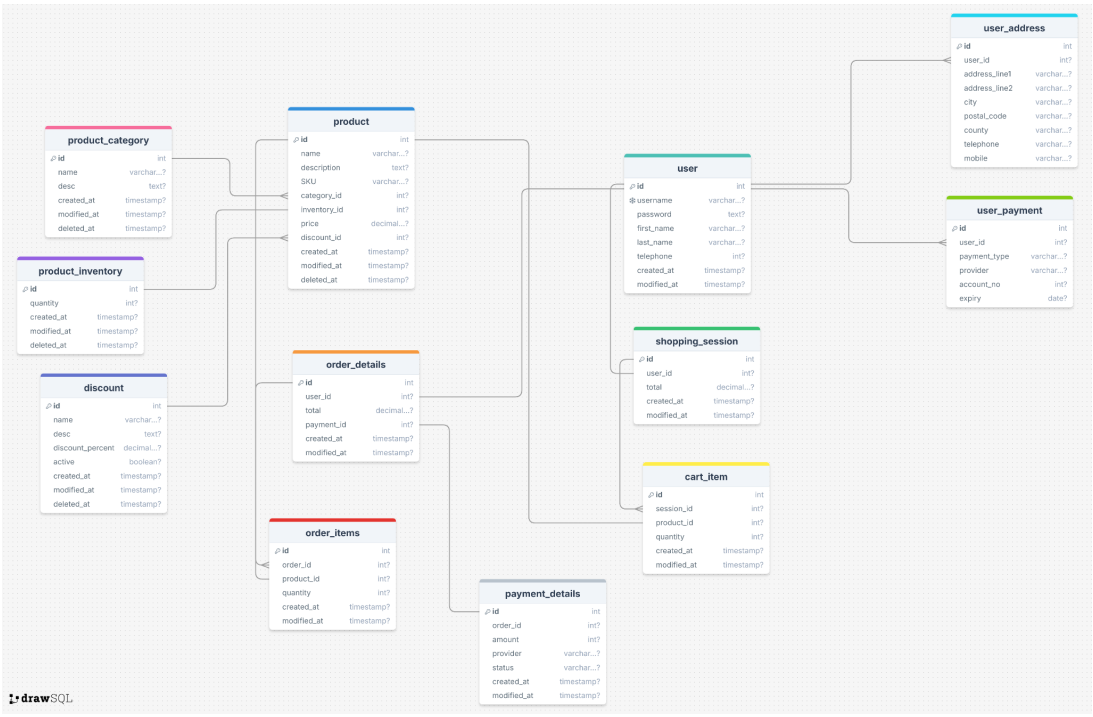


Рис. 3.2.4. ER-диаграмма базы данных: https://disk.yandex.ru/i/rktAj40xy0EJqA

Решение

Ниже представлено решение на языке Java.

Java

```java
@Entity
@Table(name = "user", schema = "public")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "usersIdSeq")
    @SequenceGenerator(name = "usersIdSeq", sequenceName = "users_id_seq", allocationSize = 1)
    @Column(name = "id")
    private Integer id;

    @Column(name = "username", length = 255)
    private String userName;

    @Column(name = "password", columnDefinition = "TEXT")
    private String password;

    @Column(name = "first_name", length = 255)
    private String firstName;

    @Column(name = "last_name", length = 255)
    private String lastName;

    @Column(name = "telephone")
    private Integer telephone;

    @Column(name = "created_at")
    @JsonProperty(access = JsonProperty.Access.READ_ONLY)
    private LocalDateTime createdAt;

    @Column(name = "modified_at")
    @JsonProperty(access = JsonProperty.Access.READ_ONLY)
    private LocalDateTime modifiedAt;

    @Column(name = "created_by")
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    @Schema(description = "The field identifies the user who created the entity. As the value of this field, you must indicate your user id in innovationcampus.ru/lms")
    private Integer createdBy;

    // Constructors
    public User() {
    }

    public User(String username, String password, String firstName, String lastName, Integer telephone, Integer createdBy, LocalDateTime createdAt, LocalDateTime modifiedAt) {
        this.userName = username;
        this.password = password;
        this.firstName = firstName;
        this.lastName = lastName;
        this.telephone = telephone;
        this.createdAt = createdAt;
        this.modifiedAt = modifiedAt;
        this.createdBy = createdBy;
    }

```

```
54      public User(String username, String password, String firstName,
        ↪  String lastName, Integer telephone, Integer createdBy) {
55          new User(username, password, firstName, lastName, telephone,
            ↪  createdBy, LocalDateTime.now(), LocalDateTime.now());
56      }
57
58      // Getters and Setters
59      public int getId() {
60          return id;
61      }
62
63      public void setId(int id) {
64          this.id = id;
65      }
66
67      public String getUserName() {
68          return userName;
69      }
70
71      public void setUserName(String userName) {
72          this.userName = userName;
73      }
74
75      public String getPassword() {
76          return password;
77      }
78
79      public void setPassword(String password) {
80          this.password = password;
81      }
82
83      public String getFirstName() {
84          return firstName;
85      }
86
87      public void setFirstName(String firstName) {
88          this.firstName = firstName;
89      }
90
91      public String getLastName() {
92          return lastName;
93      }
94
95      public void setLastName(String lastName) {
96          this.lastName = lastName;
97      }
98
99      public Integer getTelephone() {
100         return telephone;
101     }
102
103     public void setTelephone(Integer telephone) {
104         this.telephone = telephone;
105     }
106
107     public LocalDateTime getCreatedAt() {
108         return createdAt;
109     }
110
111     public void setCreatedAt(LocalDateTime createdAt) {
```

```
112         this.createdAt = createdAt;
113     }
114
115     public LocalDateTime getModifiedAt() {
116         return modifiedAt;
117     }
118
119     public void setModifiedAt(LocalDateTime modifiedAt) {
120         this.modifiedAt = modifiedAt;
121     }
122
123     public Integer getCreatedBy() {
124         return createdBy;
125     }
126
127     public void setCreatedBy(Integer createdBy) {
128         this.createdBy = createdBy;
129     }
130 }
```

Задача 3.2.1.9. Server (5 баллов)

Тема: backend-разработка.

Условие

Задача: разработать и развернуть серверное приложение, соответствующее следующему OpenAPI: https://innovationcampus.ru/store/swagger-ui.html.

Сервер должен быть доступен из сети интернет в момент проверки задания. Можно развернуть приложение на своем собственном сервере или в облачном сервисе, таком как VK Cloud или Yandex Cloud.

В качестве ответа предоставьте ссылку на развернутый сервер.

Решение

В качестве ответа участник должен предоставить ссылку на свое серверное приложение.

3.2.2. Командные задачи

В компании S контроль доступа в офис осуществляется с помощью системы контроля управления доступом (СКУД). На данный момент у каждого сотрудника компании есть карта-пропуск с NFC-меткой, а у каждой входной двери есть считыватель с обеих сторон. При поднесении карты к считывателю дверь открывается, а информация о времени входа или выхода сотрудника фиксируется в базе данных.

Администрации компании S требуется мобильное приложение как для рядовых сотрудников, так и для администрации с возможностью просмотра посещений и работой электронного пропуска в качестве временной замены обычного (при помощи

сканировании QR-кода, который находится на считывателе). Поскольку в приложении есть возможность использовать телефон как пропуск, то к данному приложению повышенные требования к безопасности всех данных, находящихся внутри него.

Важно: сервер и клиент разрабатываются на основе предоставленных заготовок проекта. Версии зависимостей и сами зависимости изменяться не должны.

Задача 3.2.2.1. Создание мобильного приложения. Клиентская часть (35 баллов)

Тема: Android-разработка.

Системные требования к приложению

- Минимальная версия ОС: Android 9.0 (API 28).
- Целевая версия ОС: Android 14 (API 34).
- Работоспособность приложения для платформ: mobile (смартфоны), tablet (планшеты).
- Поддерживаемая ориентация: портретная.
- Поддержка языков: русский, английский.
- Разрешения: доступ к интернету, камера (только при необходимости).

Техническое задание

Требуется разработать нативное мобильное приложение, которое будет содержать:

- экран авторизации,
- главный экран,
- экран с результатом сканирования QR-кода.

Экран авторизации

Экран показывается при первом входе в приложение, а также в ситуациях, когда пользователь не зарегистрировался в приложении.

Элементы, которые должны присутствовать на экране:

- Поле ввода (`id/username`), в котором пользователю необходимо ввести свой логин.
- Кнопка входа (`id/login`), после нажатия на которую пользователь авторизуется в системе.
- По умолчанию скрытое текстовое поле с ошибкой (`id/error`).

Требования к компонентам:

- В пустом поле ввода отображается подсказка, что именно требуется ввести пользователю.
- Кнопка становится неактивной, если хотя бы одно из приведенных ниже условий соблюдено:

  - ◇ поле ввода пустое;
  - ◇ количество символов менее трех;
  - ◇ логин начинается с цифры;
  - ◇ логин содержит любой другой символ, кроме цифр и латинского алфавита.
- При раскрытии клавиатуры поле ввода и кнопку должно быть видно.
- При нажатии на кнопку входа необходимо проверить, что данный пользователь существует с помощью запроса `api/<LOGIN>/auth` (подробное описание ниже).
- В случае отсутствия логина или любой другой неполадки необходимо вывести ошибку, пока пользователь не изменит текстовое поле или повторно не нажмет на кнопку.
- После нажатия на кнопку логин сохраняется и при следующем открытии приложения экран авторизации не отображается.
- После нажатия на кнопку при нажатии стрелки назад экран авторизации не должен быть показан повторно.
- Экран авторизации показывается только в случае, если пользователь не авторизован.

Главный экран

Экран содержит общую информацию о пользователе:

- Ф. И. О.
- Фото.
- Должность.
- Время последнего входа.

Элементы, которые должны присутствовать на экране:

- Текстовое поле (`id/fullname`), в котором написано имя пользователя.
- Изображение (`id/photo`), на котором отображено фото пользователя.
- Текстовое поле (`id/position`), в котором написана должность пользователя.
- Текстовое поле (`id/lastEntry`), в котором написана дата и время последнего входа пользователя.
- Кнопка (`id/logout`) для выхода пользователя из аккаунта.
- Кнопка (`id/refresh`) для обновления данных.
- Кнопка (`id/scan`) для сканирования QR-кода.
- По умолчанию скрытое текстовое поле с ошибкой (`id/error`).

Требования к компонентам:

- В случае любой ошибки необходимо скрыть все элементы, кроме текстового поля с ошибкой и кнопки обновления данных.
- Для получения данных необходимо использовать сетевой запрос `/api/<LOGIN>/info`.
- Формат даты и времени последнего входа пользователя: YYYY-MM-DDTHH:mm:ss (например: 2024-02-31T08:31). Время необходимо отображать с сервера, без поправок на часовой пояс или локальное смещение.

- При нажатии на кнопку выход все данные (если есть) пользователя должны быть очищены, а приложение должно открыть экран авторизации.
- При нажатии кнопки сканирования необходимо открыть экран сканирования QR-кода.
- При нажатии на кнопку обновления данных необходимо повторно вызывать сетевой запрос для получения актуальных данных.

Экран сканирования QR-кода

Экран позволяет отсканировать код на турникете и войти с помощью смартфона. В данном случае данный экран будет уже написан и представлен в готовом виде в заготовке. Необходимо только подписаться на его результат с помощью Result API и обработать считанные данные из QR-кода.

Данный экран нельзя модифицировать. Он поставляется в исходном виде.

Экран с результатом сканирования QR-кода

На экране необходимо вывести успешность или неуспешность входа с помощью метода QR-кода.

Элементы, которые должны присутствовать на экране:

- Текстовое поле (`id/result`), где содержится текст об успешности или неуспешности входа.
- Кнопка (`id/close`) для закрытия данного экрана.

Требования к компонентам:

- Если результат пришел пустым или со статусом **Отмена**, необходимо вывести пользователю текст «Вход был отменен/Operation was cancelled».
- Если данные пришли, следует их отправить на сервер с запросе `api/<LOGIN>/open`, добавив данные из результата и получить ответ.
- Если сервер ответил успешно, то отображаем текст: «Успешно/Success».
- Если сервер ответил любой ошибкой, то отображаем текст: «Что-то пошло не так / Something wrong».
- Кнопка закрытия всегда открывает главный экран.

Работу необходимо осуществлять в представленных проектах-заготовках (шаблонах). Шаблон для клиентской части можно найти по ссылке: https://git.sicampus.ru/Olympic/NTO-2024-Client.

Особенности оценивания

Оценивание происходит с помощью автоматической системы тестирования, которая в автоматическом режиме находит элементы и взаимодействует с ними (именно для этого у каждого элемента указан уникальный идентификатор, по которому будет производиться поиск). Каждый тест происходит с чистой установки приложения.

В случае тестирования сервера на него поочередно отправляются команды, описанные в API, и ожидаются определенные корректные ответы.

Сервер и приложение тестируются независимо.

Задача 3.2.2.2. Создание мобильного приложения. Серверная часть (35 баллов)

Темы: RESTful API, Spring Boot.

Техническое задание

Требуется разработать серверное приложение на Java (Java 11) с использованием Spring Boot, которое работает на основе протоколов TCP/IP и взаимодействует с клиентами благодаря RESTful API. Для хранения данных о сотрудниках и их посещениях будет использоваться реляционная база данных (H2). Сотрудникам не нужно регистрироваться, все данные в базе заполняются заранее. Картинки для аватаров пользователей не должны оставаться в БД, сохраняются лишь URL-адреса на ресурсы, откуда в последующем мобильное приложение загрузит соответствующий аватар.

Правила работы с проектом-заготовкой

В предоставленном проекте необходимо изучить, но никак не модифицировать, не перемещать и не удалять следующие файлы:

- `pom.xml`;
- `application.yml`;
- `data.sql`.

Кроме описанных выше файлов, в проекте уже созданы основные классы-сущности и добавлены пустые классы всех слоев. В этих классах необходимо будет написать программный код и добавить аннотации для реализации описанного задания. Наименования классов и прочий код, уже написанный в представляемом проекте, **изменять/удалять не нужно, необходимо их доработать**. Добавлять свои дополнительные классы в проект можно.

Где необходимо разместить сервер

Серверное приложение должно быть разработано и протестировано локально (**не требуется** размещать сервер удаленно и осуществлять его функционирование 24/7).

Технические требования к серверу и его ответам клиенту

Для конфигурирования сервера (его хоста/IP адреса) используйте константы из файла `Constants.kt`.

Таблица 3.2.14

Тип запроса	**Путь**	**Параметры/Тело**	**Ответы**
GET	`api/<LOGIN>/auth`	`<LOGIN>` — никнейм / имя пользователя	400 — что-то пошло не так; 401 — логина не существует или неверный; 200 — данный логин существует — можно пользоваться приложением
GET	`api/<LOGIN>/info`	`<LOGIN>` — никнейм / имя пользователя	400 — что-то пошло не так; 401 — логина не существует или неверный; 200 — OK `{ "id": 1, "login": "pivanov", "name": "Иванов Петр Федорович", "photo": "https://funnyducks.ru/upload/iblock/0cd/0cdeb7ec3ed6fddda0f90fccee05557d.jpg", "position": "Разработчик", "lastVisit": "2024-02-12T08:30:00" }`
PATCH	`api/<LOGIN>/open`	`<LOGIN>` — никнейм/имя пользователя Тело:`{"value": <CODE>}`, где: `<CODE>` — это код, полученный из экрана сканирования QR кода.	400 — что-то пошло не так; 401 — логина не существует или неверный; 200 — дверь открылась

Данные, которыми необходимо наполнить базу данных:

Таблица 3.2.15

id	**login**	**name**	**photo**	**position**	**lastVisit**
1	pivanov	Иванов Петр Федорович	https://funnyducks.ru/upload/iblock/0cd/0cdeb7ec3ed6fddda0f90fccee05557d.jpg	Разработчик	2024-02-31T08:30
2	ipetrov	Петров Иван Константинович	https://funnyducks.ru/upload/iblock/0cd/0cdeb7ec3ed6fddda0f90fccee05557d.jpg	Аналитик	2024-02-30T08:35
3	asemenov	Семенов Анатолий Анатольевич	https://funnyducks.ru/upload/iblock/0cd/0cdeb7ec3ed6fddda0f90fccee05557d.jpg	Разработчик	2024-02-31T08:31

id	login	name	photo	position	lastVisit
4	afedorov	Федоров Александр Сергеевич	https://funnyducks.ru/upload/iblock/0cd/0cdeb7ec3ed6fddda0f90fccee05557d.jpg	Тестировщик	2024-02-30T08:36

Таблица 3.2.16

id	code
1	1234567890123456789
2	9223372036854775807
3	1122334455667788990
4	998877665544332211
5	5566778899001122334

Работу необходимо осуществлять в представленных проектах-заготовках (шаблонах). Шаблон для серверной части можно найти по ссылке: https://git.sicampus.ru/Olympic/NTO-2024-Backend.

Критерии оценивания

Сервер

Проверка серверной части происходит при помощи интеграционных тестов. Участникам для выполнения задания первого этапа выдается заготовка для написания серверной части приложения. Они выполняют задание в рамках этой заготовки и присылают готовое решение.

В полученное решение добавляются интеграционные тесты и проверяется процент их успешного прохождения.

Список интеграционных тестов:

1. **Проверка подлинности для предзаполненных сотрудников.**
   Тест проверяет, что каждый предзаполненный сотрудник может успешно аутентифицироваться по логину. Для каждого сотрудника из предзаполненного списка выполняется GET-запрос на `/api/{login}/auth`. Если аутентификация прошла успешно, сервер возвращает статус `200 OK`. Тест также выводит результат запроса в консоль для наглядности. Это подтверждает, что все предустановленные данные сотрудников корректны и позволяют им входить в систему.
2. **Проверка аутентификации с неверным логином.**
   Тест проверяет, что при попытке аутентификации с логином, которого нет в базе, сервер возвращает статус `401 Unauthorized`. Таким образом, тест защищает от несанкционированного доступа, убедившись, что только существующие пользователи могут войти в систему.
3. **Проверка на работоспособность логининга не только с предзаполненными данными.**

Тест проверяет, что новый сотрудник, добавленный в базу в процессе `@BeforeEach`-метода (`testEmployee`), может успешно аутентифицироваться по логину. GET-запрос на `/api/{login}/auth` должен вернуть `200 OK`, что подтверждает возможность корректной работы приложения с динамически добавленными пользователями.

4. **Проверка получения полной информации о всех предзаполненных сотрудниках.**
   Тест проверяет, что приложение правильно возвращает полную информацию о каждом предзаполненном сотруднике по его логину. Для каждого пользователя из предзаполненного списка выполняется GET-запрос на `/api/{login}/info` и проверяется, что возвращенные данные соответствуют полям сотрудника: логин, имя, позиция, фото и дата последнего визита. Это гарантирует, что данные из базы данных корректно отдаются по запросу.
5. **Проверка полной информации без учета поля даты.**
   Тест похож на тест 4, но исключает проверку даты последнего визита, чтобы удостовериться, что остальные поля (логин, имя, позиция и фото) остаются корректными. Полезен для случаев, когда изменение формата даты может повлиять на отображение, но остальные поля остаются актуальными.
6. **Проверка на работоспособность получения полной информации не только о предзаполненных сотрудниках.**
   Тест проверяет, что новый сотрудник, добавленный в базу, отображается с полными данными. GET-запрос на `/api/login/info` возвращает статус `200 OK`, и тест проверяет все поля сотрудника (логин, имя, позиция, фото, дата последнего визита). Это подтверждает, что система обрабатывает новые записи корректно.
7. **Проверка полной информации без учета поля даты для динамически добавленного сотрудника.**
   Тест аналогичен предыдущему, но проверяет только поля логина, имени, позиции и фото без учета даты последнего визита. Это полезно для проверки основной информации о новом сотруднике без учета динамических данных, таких как дата.
8. **Проверка получения данных о несуществующем сотруднике.**
   Тест проверяет, что при запросе информации о несуществующем логине сервер возвращает статус `401 Unauthorized`. Это помогает удостовериться, что доступ к данным возможен только для существующих сотрудников, предотвращая получение информации по неверным или поддельным логинам.
9. **Проверка возможности входа динамически добавленным сотрудником в случае предоставления корректного кода.**
   Проверяет, что новый сотрудник может успешно выполнить операцию «открыть дверь», если он передает один из верных кодов из `expectedCodeList`. PATCH-запрос на `/api/{login}/open` должен возвращать `200 OK`. Это подтверждает, что сервис распознает предоставленный код как допустимый и разрешает доступ.
10. **Проверка возможности входа всеми предзаполненными сотрудниками в случае предоставления корректного кода.**
    Тест проверяет, что каждый из предзаполненных сотрудников может успешно открыть дверь, предоставив любой из корректных кодов из `expectedCodeList`. Тест проходит по всем сотрудникам и проверяет каждый код, убеждаясь, что статус ответа — `200 OK`. Это подтверждает, что система допускает

предзаполненных сотрудников с действующими кодами.

11. **Проверка обновления даты последнего визита для динамически добавленного сотрудника.**
Тест проверяет, что при успешной операции «открытие двери» для нового сотрудника дата его последнего визита обновляется. Изначально тест сохраняет дату последнего визита сотрудника, затем выполняет PATCH-запрос с правильным кодом. После успешного запроса дата последнего визита в базе должна отличаться от предыдущей, что подтверждает корректное обновление данных после успешного «входа».
12. **Проверка отсутствия возможности войти по неверному коду.**
Тест проверяет, что при передаче неверного кода (например, -1) сервер возвращает `400 Bad Request`. Это предотвращает успешную операцию «открытия двери» при передаче недействительных кодов, подтверждая корректность обработки ошибочных данных.
13. **Проверка корректности даты последнего визита.**
Тест проверяет, что при использовании неправильного кода дата последнего визита сотрудника не обновляется в базе данных. Сначала он сохраняет данные сотрудника, затем выполняет PATCH-запрос с неверным кодом, и после выполнения проверяет, что данные о сотруднике остались прежними. Это гарантирует, что при неудачном запросе данные пользователя не будут изменены.

Таблица 3.2.17

№	Описание теста	Баллы
1	Проверка логининга для предзаполненных сотрудников	3
2	Проверка аутентификации с неверным логином	2
3	Проверка на работоспособность логининга не только с предзаполненными данными	4
4	Проверка получения полной информации о всех предзаполненных сотрудниках	3
5	Проверка полной информации без учета поля даты	2
6	Проверка на работоспособность получения полной информации не только о предзаполненных сотрудниках	4
7	Проверка полной информации без учета поля даты для динамически добавленного сотрудника	2
8	Проверка получения данных о несуществующем сотруднике	2
9	Проверка возможности входа динамически добавленным сотрудником при предоставлении корректного кода	3
10	Проверка возможности входа всеми предзаполненными сотрудниками при предоставлении корректного кода	3
11	Проверка обновления даты последнего визита для динамически добавленного сотрудника	3
12	Проверка отсутствия возможности войти по неверному коду	2

13	Проверка корректности даты последнего визита (обновление при правильном коде и отсутствие обновления при неправильном)	2

Сумма за сервер — 35.

Проходной балл — 20 — прошли все тесты, для предзаполненных и хотя бы 1 для динамически добавленного.

Клиент

Проверка клиентской части происходит при помощи UI-тестов. Для выполнения задания первого этапа участникам выдается заготовка для написания клиентской части приложения, с помощью которой они выполняют задание и присылают готовое решение.

В полученное решение добавляются UI-тесты и проверяется их успешное выполнение. Каждый тест имеет разный вес, который предоставлен участникам в открытом виде. Сумма баллов приводится к стобалльной системе.

Тесты, таблица 3.2.18.

Таблица 3.2.18

Название	**Сценарий**
Авторизация	Проверяем работу с неверным логином. Проверяем работу с верным логином и последующим открытием профиля.
Выход из профиля	На профиле нажимаем кнопку выхода и проверяем переход в логин.
Кнопка обновления	Нажимаем на профиле кнопку обновить и проверяем, что данные изменились.
Сканируем неверный код	Авторизуемся и переходим в профиль, после чего сканируем QR-код с некорректным кодом. Ожидаем увидеть экран с результатом сканирования. Изменяем локаль. Проверяем, что текст изменился
Сканируем верный код	Авторизуемся и переходим в профиль, после чего сканируем QR-код с корректным кодом. Ожидаем увидеть экран с результатом сканирования. Изменяем локаль. Проверяем, что текст изменился.

Название	**Сценарий**
Отменяем сканирование QR-кода	Авторизуемся и переходим в профиль, после чего нажимаем кнопку сканирования и закрываем экран с камерой. Ожидаем увидеть экран с результатом сканирования. Изменяем локаль. Проверяем, что текст изменился.
Форматирование профиля	Переходим профиль и проверяем, обработку ошибок, а также формат вывода даты и текстовок в полях.

4. Заключительный этап

4.1. Работа наставника НТО при подготовке к этапу

На этапе подготовки к заключительному этапу НТО наставник решает две важные задачи: помощь 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/72653/enter/.

4.2.1. Информатика. 8–11 классы

Задача 4.2.1.1. ЯдерБанки (10 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: 256 Мбайт.

Условие

Android-разработчик Егор Алексеев пытался продвинуть в МИФИ стартап по разработке собственных зарядных станций и пауэрбанков «ЯдерБанки». Поскольку один из членов комиссии по обсуждению актуальности стартапа недавно попал в невыгодную ситуацию, при которой во время перехода из одного учебного корпуса в другой у него сел телефон, а зарядка осталась в первом из них, было принято решение стартовать разработку прототипа мобильного приложения для данной задачи.

В зданиях университета есть $count_d$ «ЯдерБанков», каждый из которых имеет свой *начальный уровень заряда*. Однако для клиентов важно, чтобы уровень заряда определенного «ЯдерБанка» был *не меньше* некоторого минимума к моменту, когда они им воспользуются.

Чтобы повысить уровень заряда этих «ЯдерБанков», сервис проводит $count_s$ «этапов пополнения». На каждом этапе выбираются параметры $first, last, a, b$. Затем выполняется серия действий по рядам устройств, пронумерованных подряд от 1 до N:

1. «ЯдерБанку» с номером $first$ добавляется a единиц заряда.
2. «ЯдерБанку» с номером $first + 1$ добавляется $a + b$ единиц заряда.
3. «ЯдерБанку» с номером $first + 2$ добавляется $a + 2 \cdot b$ единиц заряда.
4. «ЯдерБанку» с номером $last$ добавляется $a + (last - first) \cdot b$ единиц заряда.

Таким образом, по каждому устройству в диапазоне $[first; last]$ распределяется дополнительный заряд, который растет линейно с каждым следующим устройством.

Для заданного «ЯдерБанка» необходимо определить номер первого этапа, после которого его заряд окажется не меньше требуемой величины.

Формат входных данных

В первой строке задано целое число $count_d\,(1 \leqslant count_d \leqslant 10^5)$ — количество «ЯдерБанков».

Во второй строке заданы $count_d$ целых чисел $device_i\,(0 \leqslant device_i \leqslant 10^9)$ — начальное количество заряда в пауэрбанке с соответствующим ему индексом.

В третьей строке заданы $count_d$ целых чисел $charge_i\,(0 \leqslant charge_i \leqslant 10^9)$ — необходимое минимальное количество заряда в ЯдерБанке с соответствующим ему индексом.

В четвертой строке задано число $count_s\,(0 \leqslant count_s \leqslant 10^5)$ — количество этапов увеличения единиц заряда в «ЯдерБанках».

В следующих $count_s$ строках записаны сами этапы по порядку при помощи чисел $first, last, a, b\ (1 \leqslant first \leqslant last \leqslant count_d, 0 \leqslant a, b \leqslant 10^5)$.

Формат выходных данных

Выведите $count_d$ чисел, где для каждого «ЯдерБанка» с индексом i значение равняется `0`, если исходного заряда было достаточно уже в самом начале; номер первого этапа от одного до $count_s$, после которого уровень заряда впервые достиг или превысил требуемое значение; (-1) — если даже после всех $count_s$ пополнений необходимый уровень заряда так и не был достигнут.

Примеры

Пример №1

Стандартный ввод
5 5 4 4 2 1 7 7 4 7 7 3 1 2 2 0 2 5 1 1 3 4 2 2

Стандартный вывод
1 2 0 3 -1

Решение

В задаче требуется для каждого из n устройств с начальными зарядами a_i и целевыми уровнями b_i определить номер первой из m операций, после которых заряд этого устройства достигнет или превысит b_i. Каждая операция на отрезке $[l, r]$ добавляет линейно возрастающую последовательность (начальное значение x, приращение y). Прямое применение каждой операции к каждому элементу отрезка и проверка всех устройств после каждой операции дают время $\mathcal{O}(mn)$, что при $n, m \leqslant 10^5$ слишком медленно.

Чтобы обойти это, будем обрабатывать операции блоками по размеру S (например, $S \approx 450$). Внутри каждого блока операции накапливаются в разностных массивах с отложенными (ленивыми) обновлениями: один массив аккумулирует константную часть прибавок на отрезке, второй — коэффициент при индексе, а вспомогательный массив устраняет влияние линейного роста за пределами отрезка.

Пример программы-решения

Ниже представлено решение на языке Java.

Java

```java
import java.io.*;
import java.util.*;

public class Main {
    private static final int SZ = 450;

    static int n, m;
    static long[] a, b;
    static int[] ans;
    static long[] sumX, sumY, sub;
    static int[] l, r, x, y, num;
    static int k = 0;

    private static void doLazyUpdate() {
        for (int i = 1; i <= k; i++) {
            sumX[l[i]] += x[i];
            sumX[r[i] + 1] -= x[i];
            sumY[l[i] + 1] += y[i];
            sumY[r[i] + 1] -= y[i];
            sub[r[i] + 1] += 1L * y[i] * (r[i] - l[i]);
        }

        long add = 0;
        for (int i = 1; i <= n; i++) {
            sumX[i] += sumX[i - 1];
            long initialValue = a[i];
            a[i] += sumX[i];

            sumY[i] += sumY[i - 1];
            add += sumY[i] - sub[i];
            a[i] += add;

            if (ans[i] == -1 && a[i] >= b[i]) {
                long value = initialValue;
                for (int j = 1; j <= k; j++) {
                    if (l[j] > i || r[j] < i) continue;
                    value += x[j];
                    value += 1L * y[j] * (i - l[j]);
                    if (value >= b[i]) {
                        ans[i] = num[j];
                        break;
                    }
                }
            }

            sumX[i - 1] = 0;
            sumY[i - 1] = 0;
```

```
48              sub[i - 1] = 0;
49          }
50
51          sumX[n] = 0;
52          sumY[n] = 0;
53          sub[n] = 0;
54
55          k = 0;
56      }
57
58      public static void main(String[] args) throws IOException {
59          BufferedReader br = new BufferedReader(new
            ↪  InputStreamReader(System.in));
60          PrintWriter out = new PrintWriter(new
            ↪  BufferedOutputStream(System.out));
61          StringTokenizer st;
62
63          st = new StringTokenizer(br.readLine());
64          n = Integer.parseInt(st.nextToken());
65
66          a = new long[n + 5];
67          b = new long[n + 5];
68          ans = new int[n + 5];
69          sumX = new long[n + 5];
70          sumY = new long[n + 5];
71          sub = new long[n + 5];
72
73          st = new StringTokenizer(br.readLine());
74          for (int i = 1; i <= n; i++) {
75              a[i] = Long.parseLong(st.nextToken());
76          }
77
78          st = new StringTokenizer(br.readLine());
79          for (int i = 1; i <= n; i++) {
80              b[i] = Long.parseLong(st.nextToken());
81              ans[i] = (a[i] >= b[i] ? 0 : -1);
82          }
83
84          st = new StringTokenizer(br.readLine());
85          m = Integer.parseInt(st.nextToken());
86
87          l = new int[m + 5];
88          r = new int[m + 5];
89          x = new int[m + 5];
90          y = new int[m + 5];
91          num = new int[m + 5];
92
93          for (int i = 1; i <= m; i++) {
94              st = new StringTokenizer(br.readLine());
95              l[++k] = Integer.parseInt(st.nextToken());
96              r[k] = Integer.parseInt(st.nextToken());
97              x[k] = Integer.parseInt(st.nextToken());
98              y[k] = Integer.parseInt(st.nextToken());
99              num[k] = i;
100             if (k == SZ) {
101                 doLazyUpdate();
102             }
103         }
104         if (k > 0) {
105             doLazyUpdate();
```

```
106             }
107
108             for (int i = 1; i <= n; i++) {
109                 out.print(ans[i] + (i < n ? " " : "\n"));
110             }
111
112             out.flush();
113         }
114     }
```

Задача 4.2.1.2. Circleaning (20 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: 256 Мбайт.

Условие

Анастасия — организатор Некоторой Технологической Олимпиады по профилю «Разработка некоторых приложений». Ей было поручено организовать уборку некоторых помещений МИФИ, для чего она обратилась к услугам некоторой клининговой компании. Анастасия, недолго думая, выбрала уборку при помощи роботов-пылесосов фирмы Circleaning, которые убирают помещение по окружности с центром в точке, откуда стартовал данный пылесос, с некоторым заданным пользователем радиусом. Чтобы все успеть к открытию профиля, Анастасия заказала сразу два пылесоса для круговой уборки.

Известны координаты в двухмерном пространстве каждого из пылесосов — центр окружности их движения, а также радиус действия пылесосов. Кроме того, известно, что данные окружности пылесосов могут пересекаться. Помогите Анастасии определить общую площадь, которую приберут пылесосы.

Формат входных данных

В одной строке через пробел даны пять целых положительных чисел $first_x$, $first_y$, $second_x$, $second_y$ — координаты точки старта по Oxy первого и второго пылесоса, а также $radius$ — радиус действия для обоих пылесосов. Все пять чисел не превышают 100.

Формат выходных данных

Выведите одно вещественное число с точностью не менее трех знаков после запятой — общую площадь, которую приберут пылесосы.

Примеры

Пример №1

Стандартный ввод
`1 2 3 4 2`
Стандартный вывод
`22.84955592153876`

Решение

В этой задаче необходимо найти площадь объединения двух кругов радиуса r с центрами в точках (x_1, y_1) и (x_2, y_2). Обозначим квадрат расстояния между центрами через

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2,$$

а само расстояние через $d = \sqrt{d^2}$. Если $d \geqslant 2r$, круги не пересекаются и ответ равен $2\pi r^2$. Если $d = 0$, круги полностью совпадают и ответ равен πr^2. В остальных случаях площадь пересечения двух одинаковых кругов задается формулой

$$A_\cap = 2r^2 \arccos\left(\frac{d}{2r}\right) - \frac{d}{2}\sqrt{4r^2 - d^2},$$

и искомая площадь объединения равна

$$2\pi r^2 - A_\cap.$$

Алгоритм требует лишь константного числа операций (вычисление корня и арккосинуса) и работает за $\mathcal{O}(1)$ времени, используя $\mathcal{O}(1)$ памяти.

Пример программы-решения

Ниже представлено решение на языке Java.

Java

```java
import java.io.*;
import java.util.*;

public class light_is implements Runnable {
    private static final String FILENAME = "circleaning";

    File inFile = new File(FILENAME + ".in");
    PrintWriter out;
    BufferedReader in;

    private StringTokenizer st;

    light_is() {
        try {
            in = new BufferedReader(new FileReader(inFile));
            st = new StringTokenizer(in.readLine());
```

```
17              out = new PrintWriter(new File(FILENAME + ".out"));
18          } catch (IOException e) {
19              e.printStackTrace();
20              System.exit(239);
21          }
22      }
23
24      int nextInt() throws IOException {
25          if (!st.hasMoreTokens()) {
26              st = new StringTokenizer(in.readLine());
27          }
28          return Integer.parseInt(st.nextToken());
29      }
30
31
32      public void run() {
33          try {
34              int x1 = nextInt();
35              int y1 = nextInt();
36              int x2 = nextInt();
37              int y2 = nextInt();
38              int r = nextInt();
39              int d = (x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2);
40
41              double a;
42              if (d > 4 * r * r) {
43                  a = 0;
44              } else if (d == 0) {
45                  a = Math.PI / 2;
46              } else {
47                  a = Math.acos(0.5 * Math.sqrt(d) / r);
48              }
49
50              out.println(2 * (Math.PI * r * r - 2 * (0.5 * a * r * r -
                ↪  0.5 * (0.5 * Math.sqrt(d)) * (r * Math.sin(a)))));
                ↪
51          } catch (IOException e) {
52              e.printStackTrace();
53          } finally {
54              out.close();
55          }
56      }
57
58      public static void main(String[] args) {
59          new Thread(new light_is()).start();
60      }
61
62  }
```

Задача 4.2.1.3. Space at MEPhI (20 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: 256 Мбайт.

Условие

Во время экспедиции к дальним звездным системам ученые из НИЯУ МИФИ решили создать подробные космические карты планет и астероидных поясов. Экземпляр карты — сетка размера $2^h \times 2^h$, где каждая ячейка может быть заполнена либо космической пылью `dust`, либо оставаться пустотой `void`. В будущем эти карты будут использоваться для прокладки гиперпространственных маршрутов и постройки космических станций.

Для автоматизации процесса был написан специальный **генератор космических топонимов**. Он оперирует *шаблонами* `template`, которые описывают фрагмент будущей карты определенного размера. В самом начале доступны два базовых шаблона:

- `Dust(1)` — ячейка, занятая пылью;
- `Void(0)` — ячейка, представляющая собой пустоту.

Каждый из этих шаблонов является единичным квадратом. Далее в генераторе задаются теги вида:

```
<NewTemplate>=<SubTemplate1>, <SubTemplate2>, <SubTemplate3>,
↪  <SubTemplate4>
```

где `<SubTemplate1>, <SubTemplate2>, <SubTemplate3>, <SubTemplate4>` — ранее объявленные шаблоны размера $h \times h$. По правилам они «склеиваются» в квадраты $2h \times 2h$, образуя новый шаблон с именем `<NewTemplate>`. Схематично это можно представить как на рис. 4.2.1.

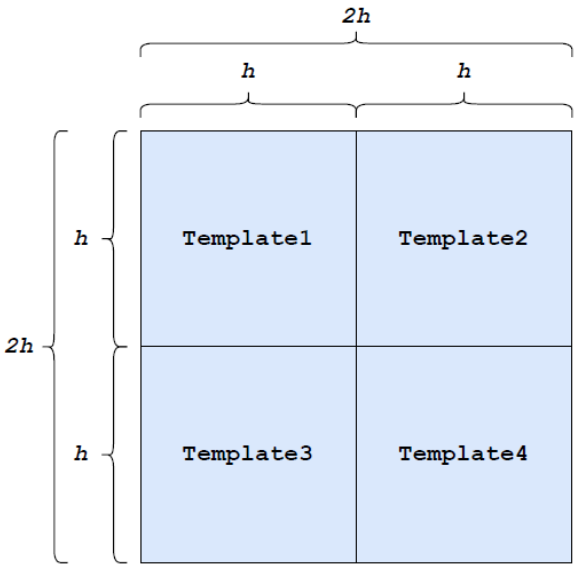


Рис. 4.2.1

В итоге после обработки всех правил получается главный шаблон `Map`, который и описывает итоговую космическую карту размером $2^h \times 2^h$. В случае, если ячейки карты имеют общую сторону, они считаются связными.

Найдите количество связных островов космической пыли на итоговой карте.

Формат входных данных

Сначала идет набор правил, задающих новые шаблоны. Каждый шаблон имеет уникальное имя, состоящее из латинских букв и цифр (прописные и строчные буквы различаются). Длина имени не превышает 20 символов.

Максимальный размер шаблона не превосходит 2^{16} для каждой из сторон, а количество шаблонов не более 200.

Завершающий шаблон, имеющий специальное имя `Map`, описывается последним. Его размер и будет размером итоговой карты.

Из этих данных формируется итоговая космическая карта.

Формат выходных данных

Требуется вывести одно число — количество связных островов космической пыли на итоговой карте.

Примеры

Пример №1

Стандартный ввод

`A=0,1,1,0`
`X=1,1,1,1`
`B=A,A,A,X`
`C=B,B,B,B`
`Map=C,C,C,C`

Стандартный вывод
`56`

Решение

В этой задаче каждый шаблон квадрата размера $2^h \times 2^h$ представляем в виде трех «метрик»: во-первых, число полностью «внутренних» островов пыли, которые не касаются границы шаблона; во-вторых, массив маркировок компонент вдоль периметра (с обходом по часовой стрелке) — каждая граница разбивается на непрерывные отрезки «пыль» или «пусто»; в-третьих, число «открытых» компонент, которые доходят до границы.

При объединении четырех квадрантов в новый шаблон, размер которого вдвое больше, их просто склеивают по периметрам: по внутренним стыкам соседних квадрантов соединяют смежные «пылевые» отрезки (DSU по меткам на границе), одновременно уменьшая число внешних компонентов, которые слились друг с другом, и добавляя к «внутренним» тем, что замкнулись полностью внутри. Таким образом, на каждом шаге сведение четырех шаблонов к новому занимает время, пропорциональное суммарному числу сегментов на их границах.

В итоге главный шаблон `Map` строится за суммарное время $\mathcal{O}\left(\sum_{\text{шаблоны}} \text{периметр}_i\right)$, что при ограничении числа правил до 200 и максимальном размере $2^{16} \times 2^{16}$ остается вполне приемлемо.

Пример программы-решения

Ниже представлено решение на языке Java.

Java

```java
import java.io.*;
import java.util.*;

public class map_as {

    Map<String, Pattern> patterns;

    class DSU {
        int n;
        int[] p;
        int[] r;

        public DSU(int n) {
            this.n = n;
            p = new int[n];
            r = new int[n];
            for (int i = 0; i < n; i++) {
                p[i] = i;
            }
        }

        public int get(int x) {
            if (p[x] != x) {
                p[x] = get(p[x]);
            }
            return p[x];
        }

        public void union(int x, int y) {
            x = get(x);
            y = get(y);
            if (r[x] == r[y]) {
                r[x]++;
            }
            if (r[x] > r[y]) {
                p[y] = x;
            } else {
                p[x] = y;
            }
        }
    }

    class Pattern {
        Pattern[] parts;

        int size;
        long inside;
        int ncol;
        int[] col;
```

```
50
51          public Pattern(int w) {
52              size = 1;
53              inside = 0;
54
55              if (w == 0) {
56                  ncol = 0;
57                  col = new int[1];
58                  col[0] = -1;
59              } else {
60                  ncol = 1;
61                  col = new int[1];
62                  col[0] = 0;
63              }
64          }
65
66          public Pattern(Pattern[] parts) {
67              this.parts = parts;
68
69              int s = parts[0].size;
70              size = s * 2;
71
72              inside = 0;
73              for (int i = 0; i < 4; i++) {
74                  inside += parts[i].inside;
75              }
76
77              int r = 4 * s - 4;
78              if (r == 0) {
79                  r = 1;
80              }
81
82              int[] ost = new int[]{3 * s - 3, 0, s - 1, 2 * s - 2};
83              int[] oen = new int[]{s % r, (2 * s - 1) % r, (3 * s - 2)
                 ↪  % r, 1 % r};
84
85              col = new int[4 * size - 4];
86              boolean[] existc = new boolean[4 * r];
87              int p = 0;
88              int q = 0;
89              for (int i = 0; i < 4 * size - 4; i++) {
90                  col[i] = parts[p].col[q];
91                  if (col[i] != -1) {
92                      col[i] += p * r;
93                      existc[col[i]] = true;
94                  }
95                  q = (q + 1) % r;
96                  if (q == oen[p]) {
97                      p = (p + 1) % 4;
98                      q = ost[p];
99                  }
100             }
101
102             DSU comps = new DSU(4 * r);
103             for (int i = 0; i < 4 * size - 4; i++) {
104                 if (i % (size - 1) == s - 1) {
105                     if (col[i] != -1 && col[(i + 1) % (4 * size - 4)]
                         ↪  != -1) {
106                         comps.union(col[i], col[(i + 1) % (4 * size -
                             ↪  4)]);
```

```
                    }
                }
            }

            for (int i = 0; i < 4; i++) {
                for (int j = oen[i] % r; j != (oen[i] + (s - 1)) % r;
                ↪  j = (j + 1) % r) {
                    int k = ((j - 2 * (j - (oen[i] - 1)) - (s - 1)) %
                    ↪  r + r) % r;
                    int c1 = parts[i].col[j];
                    int c2 = parts[(i + 1) % 4].col[k];
                    if (c1 != -1) {
                        c1 += i * r;
                        existc[c1] = true;
                    }
                    if (c2 != -1) {
                        c2 += ((i + 1) % 4) * r;
                        existc[c2] = true;
                    }
                    if (c1 != -1 && c2 != -1) {
                        comps.union(c1, c2);
                    }
                }
            }

            int[] nc = new int[4 * r];
            Arrays.fill(nc, -1);
            ncol = 0;
            for (int i = 0; i < 4 * size - 4; i++) {
                if (col[i] != -1) {
                    int c = comps.get(col[i]);
                    if (nc[c] == -1) {
                        nc[c] = ncol++;
                    }
                    col[i] = nc[c];
                }
            }

            for (int i = 0; i < 4 * r; i++) {
                if (existc[i]) {
                    int c = comps.get(i);
                    if (nc[c] == -1) {
                        nc[c] = ncol;
                        inside++;
                    }
                }
            }
        }
    }

    public void run() throws IOException {
        Scanner in = new Scanner(new File("space.in"));
        PrintWriter out = new PrintWriter(new File("space.out"));

        patterns = new HashMap<String, Pattern>();

        Pattern one = new Pattern(1);
        patterns.put("1", one);
        Pattern zero = new Pattern(0);
        patterns.put("0", zero);
```

```
165
166         while (in.hasNext()) {
167             StringTokenizer st = new StringTokenizer(in.next(), "=,");
168             String name = st.nextToken();
169             Pattern[] pp = new Pattern[4];
170             for (int i = 0; i < 4; i++) {
171                 String p = st.nextToken();
172                 pp[i] = patterns.get(p);
173             }
174             Pattern tmp = pp[2]; pp[2] = pp[3]; pp[3] = tmp;
175             Pattern pat = new Pattern(pp);
176             patterns.put(name, pat);
177         }
178
179         Pattern map = patterns.get("Map");
180         out.println(map.inside + map.ncol);
181
182         in.close();
183         out.close();
184     }
185
186     public static void main(String[] arg) throws IOException {
187         new map_as().run();
188     }
189 }
```

Задача 4.2.1.4. Энергозатраты (25 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: 256 Мбайт.

Условие

В рамках одной из некоторых программ цифровизации в некотором институте НИЯУ МИФИ начали разработку мобильного приложения, автоматизирующего передачу экспериментальных данных в профильные лаборатории и внешние запросы на некоторые научные исследования. Команда молодого Android-разработчика Никиты сильна в реализации как бэкенда, так и фронтенда, но как только касается разного рода интеграций, так все идет не по плану.

Назовем *научным исследованием* последовательность базовых экспериментов, каждый из которых имеет свою стоимость, складывающуюся из энергозатрат и стоимости обслуживания установки. При запросе исследования от сотрудников или студентов НИЯУ МИФИ применяется скидка, покрывающая только *энергозатраты* при экспериментах.

Исходное исследование можно разбить на ряд подзадач, выполняющихся менее энергозатратно. Для каждой произвольной подгруппы экспериментов стоимость энергозатрат считается как сумма затрат на все эксперименты подгруппы без $\left\lfloor \frac{count_e}{accuracy_h} \right\rfloor$ минимальных по стоимости экспериментов подгруппы, где

$count_e$ — количество экспериментов подгруппы, а $accuracy_h$ — заданный заранее уровень точности результатов экспериментов. Например, для подгруппы $[1, 1, 3, 9, 4]$ и $accuracy_h = 2$ энергозатраты составят $3 + 9 + 4 = 16$. Стоимость исследования со скидкой — минимальная сумма энергозатрат всех таких подгрупп.

Подсчитайте для заданной последовательности затрат на все эксперименты минимально возможную стоимость исследования для сотрудников МИФИ с оптимальным разбиением экспериментов на подгруппы. Исключать отдельные эксперименты или переставлять их между собой не разрешается.

Формат входных данных

В первой строке через пробел даны два целых числа $n, accuracy_h$ — количество экспериментов исследования и необходимый уровень точности результатов ($1 \leqslant n, accuracy_h \leqslant 100\,000$).

Во второй строке заданы n целых чисел $experiment_i$ ($1 \leqslant experiment_i \leqslant 10^9$) — стоимость каждого эксперимента в исследовании.

Формат выходных данных

Выведите единственное целое число — минимальную стоимость исследования со скидкой.

Примеры

Пример №1

Стандартный ввод
3 5 1 2 3

Стандартный вывод
6

Пример №2

Стандартный ввод
12 10 1 1 10 10 10 10 10 10 9 10 10 10

Стандартный вывод
92

Пример №3

Стандартный ввод
7 2 2 3 6 4 5 7 1
Стандартный вывод
17

Пример №4

Стандартный ввод
8 4 1 3 4 5 5 3 4 1
Стандартный вывод
23

Примечание.

Например, в третьем примере эксперименты можно разбить на подгруппы $[2,3]$, $[6,4,5,7]$, $[1,1]$, стоимость энергозатрат которых соответственно равна 3, 13 и 1, а итоговая стоимость исследования равна 17.

Решение

В задаче вводится одномерная последовательность затрат $a_1, \ldots, a_n$ и целый параметр k. Требуется разбить эту последовательность на непрерывные подгруппы так, чтобы суммарная стоимость

$$\sum_{\text{группа } [l..r]} \left(\sum_{i=l}^{r} a_i \; - \sum_{\substack{\text{минимальные} \\ \lfloor (r-l+1)/k \rfloor \text{ элемента}}} a_j \right)$$

была минимальна.

Опишем оптимальное разбиение динамическим программированием по префиксам. Обозначим

$$S[i] = \sum_{j=1}^{i} a_j, \qquad m(i,j) = \min_{i \leqslant t \leqslant j} a_t.$$

Тогда для $i < k$ очевидно $dp[i] = S[i]$, а для $i \geqslant k$ получаем

$$dp[i] = \min\left\{ dp[i-1] + a_i,\; dp[i-k] + \big(S[i] - S[i-k]\big) - m(i-k+1, i) \right\}.$$

Будем поддерживать на скользящем окне длины k минимум и префиксные суммы.

Пример программы-решения

Ниже представлено решение на языке Java.

Java

```java
import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new
        ↪  InputStreamReader(System.in));
        StringTokenizer st = new StringTokenizer(reader.readLine());
        int n = Integer.parseInt(st.nextToken());
        int k = Integer.parseInt(st.nextToken());
        long[] a = new long[n + 1];
        long[] dp = new long[n + 1];
        long[] mn = new long[n + 1];
        long[] sum = new long[n + 1];
        TreeMap<Long, Integer> s = new TreeMap<>();
        st = new StringTokenizer(reader.readLine());
        for (int i = 1; i <= n; i++) {
            a[i] = Long.parseLong(st.nextToken());
            s.put(a[i], s.getOrDefault(a[i], 0) + 1);
            if (i > k) {
                long x = a[i - k];
                int c = s.get(x);
                if (c == 1) {
                    s.remove(x);
                } else {
                    s.put(x, c - 1);
                }
            }
            if (i >= k) {
                mn[i] = s.firstKey();
            }
        }
        for (int i = 1; i <= n; i++) {
            dp[i] = dp[i - 1] + a[i];
            sum[i] = sum[i - 1] + a[i];
            if (i >= k) {
                long candidate = dp[i - k] + sum[i] - sum[i - k] -
                ↪  mn[i];
                if (candidate < dp[i]) {
                    dp[i] = candidate;
                }
            }
        }
        System.out.println(dp[n]);
    }
}
```

Задача 4.2.1.5. Глубокий ремонт (25 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: 256 Мбайт.

Условие

Арслан талантлив во всем. Из-за того, что в детстве и подростковом возрасте он часто ломал свои смартфоны по причине неуклюжести, в будущем он решил стать инженером микросхем. И поскольку он талантлив во всем, в качестве очередной практики своих навыков он пошел в лабораторию по ремонту и диагностике микросхем смартфонов в некотором университете.

Каждая микросхема содержит ровно $count_p$ контактных площадок (*пинов*), пронумерованных от 1 до $count_p$. Между некоторыми парами пинов $first$ и $second$ может существовать односторонняя цепочка (как провод или дорожка на плате), позволяющая протекать сигналу только в одном направлении: от $first$ к $second$. Если такая цепочка есть, то больше никаких параллельных дублей от $first$ к $second$ быть не может (то есть не может существовать двух разных проводов между той же парой пинов в одном направлении).

Назовем микросхему *безопасной* в том случае, если при уходе сигнала с любого пина вернуться на тот же пин уже невозможно. Другими словами, в схеме не должно быть замкнутых контуров: по проводам нельзя попасть из контактной площадки обратно к ней же самой.

Лаборатория собирает только безопасные микросхемы, и при этом все микросхемы должны отличаться друг от друга. Две микросхемы считаются различными, если существует хотя бы одна пара пинов $\langle first, second \rangle$, для которой в одной схеме есть цепочка от $first$ к $second$, а в другой схеме такой цепочки нет (или наоборот).

Найдите максимально возможное число таких различных безопасных микросхем, которое может собрать лаборатория, если в каждой из них ровно $count_p$ контактных площадок.

Формат входных данных

Система передачи данных к плате передает всего одно число $count_p$ — количество контактных площадок на микросхеме ($1 \leqslant counts_p \leqslant 6\,500$).

Формат выходных данных

Необходимо вычислить максимальное возможное число различных безопасных микросхем при данном количестве пинов и вывести это число по модулю $10^9 + 7$.

Примеры

Пример №1

Стандартный ввод
3

Стандартный вывод
25

Решение

Задача сводится к подсчету количества ориентированных ациклических графов (DAG) на n пронумерованных вершинах. Вершины соответствуют пинам, а ориентированное ребро $u \to v$ моделирует одностороннюю цепочку от u к v. Условие «сигнал не может вернуться на тот же пин» эквивалентно отсутствию ориентированных циклов. Две микросхемы различны, если существует пара пинов, для которой наличие пути отличается, — значит, нужно посчитать именно число DAG, а не их транзитивных замыканий.

Классическая формула включений–исключений для количества пронумерованных DAG имеет вид

$$A_0 = 1, \qquad A_n = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} 2^{k(n-k)} A_{n-k} \quad (n \geqslant 1).$$

Выбирается k «ранних» вершин, запрещаются циклы внутри них, а все ребра из этих k во внешние $n - k$ вершины задаются произвольно ($2^{k(n-k)}$ вариантов); знак $(-1)^{k-1}$ появляется из принципа включений-исключений.

Решение данной задачи сводится к эффективной реализации данного рекуррентного соотношения с условиями ограничений по времени и памяти.

Пример программы-решения

Ниже представлено решение на языке Java.

Java

```java
import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new
        ↪  InputStreamReader(System.in));
        PrintWriter out = new PrintWriter(System.out);
        int n = Integer.parseInt(br.readLine());
        n++;
        final long MOD = 1000000007;
        long[] a = new long[n + 1];
        long[] pow2 = new long[n + 1];
        long[] cpow = new long[n + 1];
        long[][] c = new long[2][n + 1];
        int f = 0;
        int ff = 1;

        pow2[0] = 1;
        a[0] = 1;
        a[1] = 1;
        for (int i = 1; i <= n; i++) {
            pow2[i] = (pow2[i - 1] * 2) % MOD;
            cpow[i] = 1;
        }
        c[1][0] = 1;
        c[1][1] = 1;
        cpow[0] = 1;
```

```
28
29          for (int i = 2; i < n; i++) {
30              c[f][0] = 1;
31              for (int j = 1; j < i; j++) {
32                  c[f][j] = c[ff][j] + c[ff][j - 1];
33                  if (c[f][j] > MOD) {
34                      c[f][j] -= MOD;
35                  }
36                  cpow[j] = (cpow[j] * pow2[j]) % MOD;
37              }
38              c[f][i] = 1;
39
40              for (int k = 1; k <= i; k++) {
41                  long tmp = a[i - k] * c[f][k] % MOD;
42                  tmp = tmp * cpow[k] % MOD;
43                  if ((k & 1) == 1) {
44                      a[i] = (a[i] + tmp) % MOD;
45                  } else {
46                      a[i] = (a[i] - tmp + MOD) % MOD;
47                  }
48              }
49              f = 1 - f;
50              ff = 1 - ff;
51          }
52
53          out.println(a[n - 1]);
54          out.flush();
55      }
56  }
```

4.2.2. Математика. 8–9 классы

Задача 4.2.2.1. (10 баллов)

Тема: делимость.

Условие

На какую цифру оканчивается число $7^{2024} + 2025^7 - 3^{2026}$?

Решение

$$7^4 = 49 \cdot 49 \equiv 1 \pmod{10};$$

$$2024 \vdots 4 \Rightarrow 7^{2024} \equiv 1 \pmod{10};$$
$$2025 \equiv 5 \pmod{10} \Rightarrow 2025^7 \equiv 5 \pmod{10};$$
$$3^{2026} = 3^{2024} \cdot 3^2 = (3^4)^{506} \cdot 9 \equiv 9 \pmod{10} \Rightarrow$$
$$\Rightarrow 7^{2024} + 2025^7 - 3^{2026} \equiv 7 \pmod{10}.$$

Ответ: 7.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно найдена и обоснована последняя цифра 7^{2024} — +3 балла.
- Верно найдена и обоснована последняя цифра 2025^7 — +1 балл.
- Верно найдена и обоснована последняя цифра 3^{2026} — +3 балла.
- Верный ответ — +3 балла.

Задача 4.2.2.2. (20 баллов)

Тема: решение уравнений в целых числах.

Условие

Решите уравнение в целых числах $x^2 = 2 \cdot y! - 1$.
Факториал — функция, определенная на множестве неотрицательных целых чисел. Можно задать рекуррентной формулой:

$$n! = \begin{cases} n \cdot (n-1)! & \text{при } n > 0, \\ 1 & \text{при } n = 0. \end{cases}$$

Решение

1. $y = 0$ или $y = 1 \Rightarrow y! = 1 \Rightarrow x^2 = 1 \Leftrightarrow x = \pm 1$.

2. $y = 2 \Rightarrow \begin{cases} x^2 = 3, \\ x \in \mathbb{Z}. \end{cases} \Rightarrow x \in \varnothing.$

3. $y = 3 \Rightarrow \begin{cases} x^2 = 11, \\ x \in \mathbb{Z}. \end{cases} \Rightarrow x \in \varnothing.$

4. $y \geqslant 3 \Rightarrow y! \vdots 3$, $x^2 + 1 = 2 \cdot y!$. Но x^2 при делении на 3 дает только остатки 0 или 1, следовательно, $x^2 + 1$ не делится на 3.

Ответ: $(1;0)$, $(1;1)$, $(-1;0)$, $(-1;1)$.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно найдены и обоснованы все пары чисел — $3 + 3 + 3 + 3 = 12$ баллов.
- Верно доказано, что других решений нет — $+8$ баллов.

Задача 4.2.2.3. (20 баллов)

Тема: комбинаторика.

Условие

Дано бесконечное множество отрезков с длинами 1, 5, 6 и 7 единиц. Из них составили треугольники.

1. Сколько всего различных видов треугольников получилось? Каждого вида взяли по одному. Найдите сумму их периметров.
2. Сколько среди этих треугольников равнобедренных?

Решение

1. 14 различных треугольников:
   - 1, 1, 1;
   - 1, 5, 5;
   - 1, 6, 6;
   - 1, 7, 7;
   - 5, 5, 5;
   - 5, 5, 6;
   - 5, 5, 7;
   - 5, 6, 6;
   - 5, 7, 7;
   - 5, 6, 7;
   - 6, 6, 6;
   - 6, 6, 7;
   - 6, 7, 7;

- 7, 7, 7.

$\sum_{i=1}^{14} P_i = 222$.

2. Девять равнобедренных треугольников:
   - 1, 5, 5;
   - 1, 6, 6;
   - 1, 7, 7;
   - 5, 5, 6;
   - 5, 5, 7;
   - 5, 6, 6;
   - 5, 7, 7;
   - 6, 6, 7;
   - 6, 7, 7.

   Четыре равносторонних треугольника:
   - 1, 1, 1;
   - 5, 5, 5;
   - 6, 6, 6;
   - 7, 7, 7.

Ответ:

1. 14; 222;
2. 9, 13.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Пункт 1: верно и обоснованно найдено количество различных треугольников — +8 баллов.
- Пункт 1: верно и обоснованно найдена сумма их периметров — +8 баллов.
- Пункт 2: верно найдено количество равнобедренных треугольников — +4 балла.

Задача 4.2.2.4. (20 баллов)

Тема: планиметрия.

Условие

Дана равнобедренная трапеция $ABCD$ с основаниями $BC = 8$ и $AD = 24$. Диагонали BD и AC пересекаются в точке O, точка P — середина стороны AB, $PO = 10$. Найдите площадь трапеции $ABCD$.

Решение

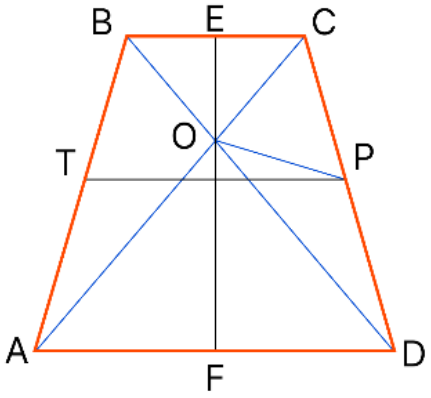


Рис. 4.2.2

1. Пусть TP — средняя линия трапеции, EF — высота трапеции, проходящая через точку O. $EF \cap TP = K$.
2. $TP = \dfrac{BC + AD}{2} = 16$, $KP = 8$.
3. $\triangle OPK$ прямоугольный, $OK = 6$.
4. $\triangle BEO \sim \triangle DFO$ по двум углам: $\angle BOE = \angle DOF$ (вертикальные), $\angle EBO = \angle FDO$ (внутренние накрест лежащие при $AD \parallel BC$ и секущей BD). Отсюда следует: $\dfrac{BE}{FD} = \dfrac{1}{3} = \dfrac{EO}{OF}$.
5. $EO = \dfrac{1}{4}EF$, $EK = \dfrac{1}{2}EF \Rightarrow OK = \dfrac{1}{4}EF = 6 \Rightarrow EF = 24$.
6. $S_{ABCD} = 16 \cdot 24 = 384$.

Ответ: 384.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно и обоснованно найдена длина катета KP — +6 баллов.
- Верно и обоснованно найдена длина катета KP — +2 балла.
- Верно и обоснованно найдена высота трапеции — +6 баллов.
- Верно найдена площадь трапеции — +6 баллов.

Задача 4.2.2.5. (30 баллов)

Тема: решение алгебраических уравнений.

Условие

Решите уравнение $\dfrac{x^3+9x^2+27x+27}{\sqrt{x+5}} = x+5$.

Решение

1. $\dfrac{(x+3)^3}{\sqrt{x+5}} = x+5 \Rightarrow \begin{cases}(x+3)^3 = (\sqrt{x+5})^3,\\ x \neq -5.\end{cases}$
2. $f(t) = t^3$ — монотонная функция $\Rightarrow x+3 = \sqrt{x+5}$.
3. $\begin{cases}x^2+6x+9 = x+5,\\ x > -3.\end{cases} \Rightarrow \begin{cases}x^2+5x+4 = 0,\\ x > -3.\end{cases} \Rightarrow \begin{cases}x = -4 \text{ или } x = -1,\\ x > -3.\end{cases} \Rightarrow$
$\Rightarrow x = -1.$

Ответ: -1.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно и обоснованно пришел к уравнению $x+3 = \sqrt{x+5}$ — +10+7 баллов.
- Верный равносильный переход — +5 баллов.
- Верно решено квадратное уравнение — +5 баллов.
- Верно отобран корень — +3 балла.

4.2.3. Математика. 10–11 классы

Задача 4.2.3.1. (10 баллов)

Тема: делимость.

Условие

На какую цифру оканчивается число $7^{2025} - 2024^7 - 12^{2026}$?

Решение

$$7^4 = 49 \cdot 49 \equiv 1 \pmod{10};$$
$$7^{2025} = (7^4)^{506} \cdot 7 \equiv 7 \pmod{10};$$
$$2024^7 \equiv 4^6 \cdot 4 \equiv 6 \cdot 4 \equiv 4 \pmod{10};$$
$$12^{2026} = (3^4)^{506} \cdot 3^2 \cdot (4^2)^{1013} \equiv 1 \cdot 9 \cdot 6 \equiv 4 \pmod{10};$$
$$7^{2025} - 2024^7 - 12^{2026} < 0;$$
$$7^{2025} - 2024^7 - 12^{2026} \equiv 7 - 4 - 4 = -1.$$

Число оканчивается на 1.

Ответ: 1.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно найдена и обоснована последняя цифра 7^{2025} — +2 балла.
- Верно найдена и обоснована последняя цифра 2024^7 — +3 балла.
- Верно найдена и обоснована последняя цифра 12^{2026} — +3 балла.
- Верный ответ — +2 балла.

Задача 4.2.3.2. (20 баллов)

Тема: решение уравнений в целых числах.

Условие

Решите уравнение в целых числах $x^2 = y! + 3$.

Факториал — функция, определенная на множестве неотрицательных целых чисел. Можно задать рекуррентной формулой:

$$n! = \begin{cases} n \cdot (n-1)! & \text{при } n > 0, \\ 1 & \text{при } n = 0. \end{cases}$$

Решение

1. $y = 0$ или $y = 1 \Rightarrow y! = 1 \Rightarrow x^2 = 4 \Leftrightarrow x = \pm 2$.
2. $y = 2 \Rightarrow \begin{cases} x^2 = 5 \\ x \in \mathbb{Z} \end{cases} \Rightarrow x \in \varnothing$.
3. $y = 3 \Rightarrow \begin{cases} x^2 = 9 \\ x \in \mathbb{Z} \end{cases} \Rightarrow x = \pm 3$.
4. $y \geqslant 4 \Rightarrow y! \vdots 4$, $y! + 3 \equiv 3 \pmod 4$. Но x^2 при делении на 4 дает только остатки 0 или 1.

Ответ: $(\pm 2;\ 0)$, $(\pm 2;\ 1)$, $(\pm 3;\ 3)$.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно найдены и обоснованы все пары чисел — 2+2+2+2+2+2=12 баллов.
- Верно доказано, что других решений нет — +8 баллов.

Задача 4.2.3.3. (20 баллов)

Темы: комбинаторика, теория графов.

Условие

Дана 2025-угольная призма.

1. Можно ли раскрасить все вершины призмы в 3 цвета так, чтобы концы каждого ребра были разных цветов и на каждой грани были вершины всех трех цветов?
2. Можно ли раскрасить все ребра призмы в 3 цвета так, чтобы в каждой вершине сходились ребра всех трех цветов и на каждой грани были ребра всех трех цветов?

Решение

1. $2025 \vdots 3$.
   Занумеруем все вершины на одном из оснований 1, 2, 3, 1, 2, 3. . . и в том же направлении — на другом со сдвигом на одно ребро.

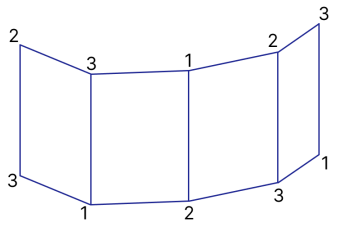


Рис. 4.2.3

2. Достаточно привести подтверждающий пример, см. рис. 4.2.4.

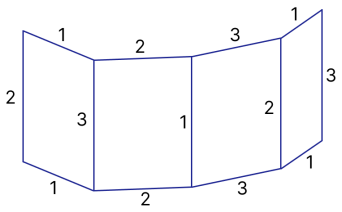


Рис. 4.2.4

Ответ:

1. да;
2. да.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Пункт 1: 10 баллов.
- Пункт 2: 10 баллов.
- Достаточно привести подтверждающий пример для получения полного балла.

Задача 4.2.3.4. (25 баллов)

Тема: комбинаторика.

Условие

Дано бесконечное множество отрезков с длинами 2, 4, 5 и 6 единиц. Из них составили треугольники.

1. Сколько всего различных видов треугольников получилось? Взяли по одному треугольнику каждого вида. Найдите сумму их периметров.
2. Из выбранных треугольников взяли один. Какова вероятность взять равнобедренный треугольник?

Решение

1. 16 различных треугольников:
   - 2, 2, 2;
   - 2, 4, 4;
   - 2, 4, 5;
   - 2, 5, 5;
   - 2, 5, 6;
   - 2, 6, 6;
   - 4, 4, 4;
   - 4, 4, 5;
   - 4, 4, 6;
   - 4, 5, 5;
   - 4, 5, 6;
   - 4, 6, 6;
   - 5, 5, 5;
   - 5, 5, 6;
   - 5, 6, 6;
   - 6, 6, 6.

   $\sum_{i=1}^{16} P_i = 216$.
2. 9 равнобедренных треугольников:
   - 2, 4, 4;
   - 2, 5, 5;
   - 2, 6, 6;
   - 4, 4, 5;
   - 4, 4, 6;
   - 4, 5, 5;
   - 4, 6, 6;
   - 5, 5, 6;
   - 5, 6, 6.

   $P = \dfrac{9}{16}$.

4 равносторонних треугольника:

- 2, 2, 2;
- 4, 4, 4;
- 5, 5, 5;
- 6, 6, 6.

$P = \dfrac{13}{16}$.

Ответ:

1. 16; 216;
2. $\dfrac{9}{16}$, $\dfrac{13}{16}$.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Пункт 1: верно и обоснованно найдено количество различных треугольников — +10 баллов.
- Пункт 1: верно и обоснованно найдена сумма их периметров — +10 баллов.
- Пункт 2: верно найдено количество равнобедренных треугольников — +3 балла.
- Пункт 2: верно найдена искомая вероятность — +2 балла.

Задача 4.2.3.5. (25 баллов)

Тема: планиметрия.

Условие

Дана трапеция $ABCD$. На основании AD взята точка K так, что $\dfrac{AK}{AD} = \dfrac{1}{3}$. Точку K соединили с вершинами верхнего основания трапеции BC. Отрезки BK и KC пересекли диагонали трапеции AC и BD в точках M и T соответственно. Найдите площадь треугольника BTC, если известно, что площадь трапеции равна 48, а M — середина отрезка AC.

Решение

1. $AM = MC$, $\angle AMK = \angle CMB$ (вертикальные), $\angle BCM = \angle MAK$ (внутренние накрест лежащие при $AD \parallel BC$ и секущей AC).
   Отсюда: $\triangle AMK = \triangle CMB$ по стороне и двум прилежащим углам. Значит, $AK = BC \Rightarrow \dfrac{BC}{AD} = \dfrac{1}{3}$.
2. $\dfrac{S_{BCD}}{S_{ABD}} = \dfrac{1}{3}$, $S_{BCD} = \dfrac{1}{4} S_{ABCD} = 12$.
3. $\triangle BTC \sim \triangle DTK$ по двум углам: $\angle BTC = \angle DTK$ (вертикальные), $\angle BCT = \angle TKD$ (внутренние накрест лежащие при $AD \parallel BC$ и секущей KC).

Отсюда следует, что $\dfrac{BT}{TC} = \dfrac{BC}{KD} = \dfrac{1}{2}$, а значит, $S_{BTC} = \dfrac{1}{3} S_{BCD} = 4$.

Ответ: 4.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Если ответ верный, но решение недостаточно обосновано — от 10 до 20 баллов.

4.3. Инженерный тур

4.3.1. Общая информация

Задание инженерного тура заключительного этапа максимально приближено к реальной разработке мобильных приложений, предоставляя участникам возможность испытать свои навыки в этой сфере.

4.3.2. Легенда задачи

В компании S контроль доступа в офис осуществляется с помощью СКУД (системы контроля управления доступом). На данный момент у каждого сотрудника есть карта-пропуск с NFC-меткой, а у каждой входной двери — считыватель с обеих ее сторон. При поднесении карты к считывателю дверь открывается, и информация о времени входа или выхода сотрудника фиксируется в базе данных.

Администрации компании S требуется мобильное приложение (как для рядовых сотрудников, так и для администрации) с возможностью просмотра посещений и работой электронного пропуска как временной замены обычного (при помощи сканировании QR-кода, который находится на считывателе). В приложении есть возможность использовать телефон как пропуск, поэтому к безопасности данных, находящихся внутри него, предъявляются повышенные требования.

Сервер и клиент разрабатываются на основе командного решения второго этапа.

4.3.3. Требования к команде и компетенциям участников

Количество участников в команде: 2–3 человека.

Роли:

- **Android-разработчик**: отвечает за разработку клиентской части программного продукта (Android-приложение), которое будет взаимодействовать с сервером. Необходимые hard-скиллы:
    - ◇ Язык программирования: знание Java и/или Kotlin как основных языков для разработки приложений под Android.
    - ◇ Android SDK: понимание и умение работать с Android Software Development Kit, включая библиотеки, инструменты и различные API, с помощью которых создаются нативные Android-приложения.
    - ◇ Инструменты разработки: умение использовать Android Studio и Gradle для разработки, сборки и развертывания приложений.
    - ◇ Архитектура приложений: знание архитектурных паттернов (MV*, Clean

Architecture) для организации кода и улучшения его читаемости и поддерживаемости.

- ◇ Работа с данными: опыт работы с SQLite, библиотекой Room или другими базами данных, а также навыки работы с RESTful API и JSON для обмена данными с сервером.
- ◇ Тестирование: опыт написания unit-тестов с использованием соответствующих фреймворков для обеспечения надежности и стабильности кода.

- **UI/UX-дизайнер:** проектирование интерфейса, создающего комфортный пользовательский опыт Android-приложения. Необходимые hard-скиллы (роли «Android-разработчик» и «UI/UX-дизайнер» могут совмещаться):
  - ◇ Инструменты дизайна: умение работать с инструментами для создания дизайна и прототипов, такими как Figma или Sketch.
  - ◇ Платформенные гайды: знание Android Material Design Guidelines или One UI, что позволяет создавать интерфейсы, соответствующие стандартам и рекомендациям платформы.
  - ◇ Верстка интерфейсов: владение XML или библиотекой JetPack Compose для создания макетов пользовательского интерфейса и определения ресурсов.
  - ◇ Инструменты для создания графики: владение Gimp и Inkscape или другими аналогичными растровыми и векторными редакторами для создания графических элементов и иконок, необходимых для приложений.
- **Бэкенд-разработчик:** реализация серверной части программного продукта, которая будет предоставлять необходимую информацию Android-приложению. Необходимые hard-скиллы:
  - ◇ Языки программирования: владение языками программирования, такими как Java, Kotlin или другими популярными языками, которые используются для написания серверной логики.
  - ◇ Базы данных: знание реляционных баз данных (например, PostgreSQL, MySQL) и нереляционных баз данных (например, MongoDB), а также умение создавать запросы.
  - ◇ API-разработка: опыт разработки и интеграции RESTful, включая создание и документирование эндпоинтов, а также обеспечение безопасности и производительности.
  - ◇ Фреймворки и библиотеки: знание Spring Boot, который упрощает разработку серверной части и управление зависимостями в Spring-приложениях.
  - ◇ Архитектура приложений: понимание принципов архитектуры приложений, включая микросервисную, модульную и монолитную архитектуру. Способность проектировать системы с учетом масштабируемости и отказоустойчивости. Знание архитектурных паттернов.
  - ◇ Тестирование: опыт написания unit-тестов с использованием соответствующих фреймворков для проверки корректности серверной логики.
- **Тимлид** (один из членов команды): координация работы команды, а также коммуницирование с организаторами профиля. Необходимые hard-скиллы:
  - ◇ Обзорное представление обо всем, что описано выше.

- ◇ Управление проектами и Agile: опыт работы с инструментами управления проектами (JIRA, Kaiten). Умение знать и применять Agile-методологии (Scrum, Kanban).

В команде не может быть человека ТОЛЬКО с ролью тимлида, эта роль всегда совмещается с перечисленными выше ролями Android-разработчика, UI/UX-дизайнера, бэкенд-разработчика.

Рекомендации по взаимодействию в команде можно найти в мастер-классе Samsung Android Bootcamp 2025: https://vkvideo.ru/video-226950057_456239557?list=ln-MJ9lFnFzNXnSU4jA4n (для просмотра видео необходимо подписаться).

4.3.4. Оборудование и программное обеспечение

Таблица 4.3.1

Наименование	**Описание**
JDK 11	Java Development Kit — все, что нужно для разработки на языке Java
JetBrains IntelliJ IDEA Community Edition Ladybug Feature Drop | 2024.2.2	Среда разработки на языке Java; для разработки бэкэнда
Android Studio Ladybug Feature Drop | 2024.2.2 (minimum sdk 28, target sdk 34) и встроенным эмулятором	Среда разработки для Android; для разработки фронтэнда
Postman 11.31.4	Приложение для работы с API
Insomnia 10.3.0	Приложение для работы с API
MySQL 9.2.0	СУБД
PostgreSQL 17.2	СУБД
DBeaver Community 24.3.4	Приложение с графическим интерфейсом для работы с СУБД
Docker Desktop	Приложение для контейнеризации
Git	Система контроля версий

4.3.5. Описание задачи

Создание мобильного приложения. Клиентская часть

Тема: Android-разработка.

Системные требования к приложению:

- Минимальная версия ОС: Android 9.0 (API 28).
- Целевая версия ОС: Android 14 (API 34).
- Работоспособность приложения для платформ: mobile (смартфоны), tablet

(планшеты).

- Поддерживаемая ориентация: портретная, альбомная.
- Поддержка языков: русский, английский.
- Разрешения: доступ к интернету, камера (только при необходимости).

Техническое задание

Требуется доработать нативное мобильное приложение, которое было выполнено командой на втором этапе Олимпиады. За основу следует взять приложение предыдущего этапа, доработать его и добавить следующий функционал в экраны:

- Экран авторизации: доработать авторизацию, добавив поле пароля; при успешной аутентификации необходимо реализовать Basic-авторизацию.
- Главный экран: дополнить прокручиваемым списком (на основе компонента RecyclerView), в каждом элементе которого содержится информация:
  - ◇ время сканирования считывателя,
  - ◇ идентификатор или название считывателя,
  - ◇ тип прохода (карта или вход через смартфон).
- Экран администратора: разработать новый экран, доступ к которому будет только у пользователей с правами администратора; на нем должно содержаться поле ввода логина сотрудника (после успешного ввода он может просмотреть информацию о сотруднике), а также кнопка блокировки и разблокировки пропуска с помощью смартфона сотрудника.

Создание мобильного приложения. Серверная часть

Темы: RESTful API, Spring Boot.

Техническое задание

Доработать сервер. Реализовать ролевую модель, разделив права доступа к эндпоинтам. Данные обо всех сотрудниках, с детализацией их посещений офиса, должны быть доступны только пользователям с ролью «Администратор». Остальные эндпоинты, кроме аутентификации, должны быть у всех авторизованных пользователей.

Сведения, необходимые для аутентификации и авторизации, сохраняются в базе данных. Пароль обязательно хранится в зашифрованном виде. Для реализации аутентификации и авторизации рекомендовано использовать Spring Security.

Где необходимо разместить сервер

Серверное приложение должно быть развернуто и протестировано локально на одном из рабочих мест участников команды.

4.3.6. Система оценивания

Оценка решения задачи проводится в два этапа:

- техническая экспертиза;
- защита перед жюри.

Оба этапа проходят по завершении работы над заданием. Техническая экспертиза заочно оценивает код представленного решения. На очной защите оценивается работоспособность приложения и его соответствие техническому заданию.

Критерии технической экспертизы представлены ниже.

Критерии Backend

Таблица 4.3.2

	Название	**Описание**	**Балл**
K1	Нормализация данных в БД	Оценивается структура базы данных, ее соответствие принципам нормализации. Проверяется, избегаются ли дублирующиеся данные, эффективно ли используются связи между таблицами, а также грамотно ли выбраны типы данных для столбцов: • использование 1NF — 4 балла, • использование 2NF — 6 балла, • использование 3NF — 6 балла, • оптимизация индексов и грамотный выбор типов данных — 4 балла.	20
K2	Архитектура микросервиса	Оценивается модульность и согласованность компонентов микросервиса. Проверяется соблюдение принципов SOLID, четкое разделение слоев (Controller, Service, Repository) и использование Dependency Injection. Учитывается, насколько удобно расширять и модифицировать микросервис, а также соответствие архитектуры задачам. Дополнительным плюсом будет использование паттернов проектирования, таких как DTO, Factory, Builder и т. д.: • четкое разделение слоев (Controller, Service, Repository) — 6 баллов, • применение принципов SOLID — 6 баллов, • использование Dependency Injection — 4 балла, • расширяемость и модифицируемость — 4 балла, • использование паттернов проектирования — 4 балла.	24

	Название	**Описание**	**Балл**
К3	Обработка исключений	Оценивается, насколько эффективно и полно обрабатываются возможные исключения. Проверяется, есть ли централизованный механизм обработки ошибок (например, `@ControllerAdvice` и `@ExceptionHandler`), предоставление информативных сообщений об ошибках клиенту, а также защита системы от некорректного ввода или неожиданных сбоев: • наличие централизованного механизма обработки ошибок — 6 баллов, • информативные сообщения об ошибках — 4 балла, • защита от некорректного ввода и сбоев — 4 балла, • работа с кастомными исключениями — 2 балла.	16
К4	Аутентификация и авторизация	Оценивается реализация безопасной аутентификации и авторизации, включая использование современных протоколов (например, JWT, OAuth2 или Basic). Проверяется корректность настройки цепочки фильтров безопасности (Spring Security): • реализация аутентификации с использованием современных протоколов (JWT, OAuth2, Basic) — 8 баллов, • настройка фильтров безопасности в Spring Security — 8 баллов.	16
К5	Реализация ролевой модели	Оценивается, насколько гибко и полноценно реализована ролевая модель в системе. Проверяется, существует ли разграничение прав доступа к ресурсам в зависимости от ролей, корректность проверки привилегий и удобство добавления новых ролей (плюсом считается наличие документации, описывающей права каждой роли, и тестов): • разграничение доступа по ролям — 6 баллов, • роли вынесены в отдельную таблицу, реализована связь многие ко многим (или один ко многим) — 4 балла, • наличие документации и тестов, подтверждающих корректность ролевой модели — 4 балла.	14
К6	Оптимизация	Оценивается оптимизация запросов к БД: • решена проблема N+1 — 4 балла, • реализована пагинация — 4 балла.	8
К7	Тесты	Более 30% логики покрыто unit-тестами. Подключены механизмы анализа покрытия кода тестами (например: JaCoCo): • покрытие логики тестами более 30% — 1 балл, • использование библиотек по анализу покрытия кода тестами — 1 балл.	2

	Название	**Описание**	**Балл**
К8	Объем самостоятельной работы	Оценка самостоятельной работы каждого участника	0–100%

Итоговый балл рассчитывается по формуле:

$$(K1 + K2 + K3 + K4 + K5 + K6 + K7) \times K8.$$

Максимальный балл — 100.

Критерии Frontend

Таблица 4.3.3

	Название	**Описание**	**Балл**
К1	Наличие всех базовых элементов интерфейса	Приложение должно включать следующие интерфейсы: • Авторизация (3). • Главный экран (3). • Экран сканирования QR кода (1). • Экран с результатом сканирования QR кода (3).	10
К2	Основной функционал	• Реализована авторизация с паролем (12): ◇ используется для всех запросов в приложении (6); ◇ выполнена обработка при устаревании данных авторизации (4); ◇ используется ли шифрование при локальном хранении данных авторизации (2). • Реализован список последних входов (12): ◇ при смене конфигурации список сохраняет свою позицию (2); ◇ список загружается постранично (4); ◇ список содержит информации не меньше, чем описано в требованиях (3); ◇ элементы списка переиспользуются (3). • Реализован экран администратора (16): ◇ поиск сотрудника по логину (2); ◇ возможность заблокировать и разблокировать пропуск (2); ◇ просмотр истории пользователя (6); ◇ визуальное скрытие раздела для пользователей без доступа (4); ◇ обработка состояния блокировки пропуска при считывании QR (2).	40

	Название	**Описание**	**Балл**
К3	Наличие дополнительного функционала (если К2 = 0 — не оценивается)	Выбрать одно: • дополнительного функционала, не оговоренного заданием, нет — 0; • присутствует несложный дополнительный функционал или улучшение текущего — 4; • присутствует функционал, добавляющий дополнительные сценарии использования приложения — 6; • выполнены значительные качественные доработки и внедрен совершенно новый сценарий использования — 10.	10
К4	UX-приложения	В приложении должны быть реализованы следующие механики: • Авторизация. Вывод информации об ошибке (2). • Главный экран. Обработка состояний отсутствия интернета с возможностью повторить попытку (1,5). • Экран обработки результата. Обработка состояния отсутствия интернета с возможностью повторить попытку (1,5).	5
К5	Архитектура	• Код разделен на три слоя: представление, домен, данные (2). • Правильно выстроены зависимости между слоями (1). • Для более 75% кода правильно выбран слой (1). • Для представления выбран MV* паттерн (2).	6
К6	Код	• Код декомпозирован на классы и методы (3). • Более 90% кода не дублируется (3). • Кодовая база не содержит устаревших библиотек или вызовов API (2).	8
К7	Тесты	Более 30% логики покрыто unit-тестами	7
К8	Использование технологий	• Использование нестандартного обмена данными (например, GraphQL) (1). • Использование локального хранилища данных на основе ORM (1). • Использование библиотек многопоточности и использование ее для всех длительных операций (1). • Использование альтернативных способов считывания меток (NFC) (1). • Критерий субъективный: можно добавить балл за библиотеку/технологию, которая достойна быть дополнительно оценена.	4
К9	Работоспособность приложения	Приложение работает и запускается без сбоев на всех поддерживаемых версиях Android: если да — полный балл, иначе — 0.	10

	Название	**Описание**	**Балл**
K10	Объем самостоятельной работы	Оценка самостоятельной работы каждого участника	0–100%

Формула для расчета итогового балла:

$$(K1 + K2 + \cdots + K8 + K9) \times K10.$$

Максимальный балл — 100.

Критерии UI/UX

Таблица 4.3.4

	Название	**Описание**	**Балл**
K1	Соответствие UI-гайдлайнам	Экранные формы приложения выполнены в соответствии с гайдлайнами Material Design 3 / One UI: • Интерфейс строго соответствует гайдлайнам (15). • Есть небольшие отклонения, но общая концепция соблюдена (10). • Применены базовые принципы, но гайдлайны нарушены (5). • Интерфейс полностью игнорирует гайдлайны (0).	15
K2	Основной функционал	Экранные формы полностью реализуют заявленный функционал: • Авторизация (2). • Главный экран (2). • Экран сканирования QR кода (2). • Экран с результатом сканирования QR кода (2). • Экран администратора (2).	10
K3	Единая визуальная стилистика и согласованность элементов	Экранные формы приложения: • выполнены в едином стиле, элементы выглядят согласованно (5); • выполнены с небольшими помарками (разные шрифты, отступы, несоответствие цветов) (3); • выполнены хаотично, элементы не связаны (0).	5
K4	Качество проработки цветовой палитры и типографики	Выбрать одно: • Цвета, шрифты и размеры текста грамотно подобраны (5). • Есть небольшие ошибки (нечитабельность, плохой контраст) (3). • Полностью неудачное сочетание цветов и шрифтов (0).	5

	Название	**Описание**	**Балл**
К5	Использование графики и иллюстраций	Выбрать одно: • Используются качественные иконки, иллюстрации, изображения в правильном формате (иконографика — векторная, svg.: фото — растровое, webp) (5). • Используются иконки/иллюстрации/изображения в неправильном формате (3). • Дизайн без иконок/иллюстраций (0).	5
К6	Логичность и интуитивность навигации	Выбрать одно: • Интерфейс интуитивно понятен, и пользователю не требуются инструкции для его освоения (10). • Есть небольшие сложности в навигации (5). • Навигация сложная, непонятная (0).	10
К7	Проработанность пользовательских сценариев	Выбрать одно: • Все важные сценарии продуманы и протестированы (5). • Частично продуманы, но есть узкие места (3). • Некоторые сценарии не проработаны (0).	5
К8	Удобство ввода данных и обработки ошибок	Выбрать одно: • Формы ввода удобные, ошибки обрабатываются понятно (5). • Ошибки отображаются, но не всегда корректно или представлены в непонятном виде (3). • Ошибки игнорируются, пользователю сложно вводить данные (0).	5
К9	Адаптивность	Выбрать одно: • Интерфейс корректно отображается на разных устройствах и экранах, включая поворот (портретная и альбомная ориентации) (10). • Есть незначительные проблемы с адаптацией (например, мелкие несоответствия в размерах элементов) (5). • Интерфейс ломается при изменении разрешения экрана или повороте устройства (0).	10
К10	Оптимизация производительности	Выбрать одно: • Верстка оптимизирована. Минимизирована вложенность View (XML) или композиций (Compose) (5). • Есть незначительные проблемы с оптимизацией: например, небольшая избыточная вложенность (3). • Верстка не оптимизирована (0).	5

	Название	**Описание**	**Балл**
K11	Наличие и качество макета	Выбрать одно: • Макет предоставлен, полностью детализирован (включая размеры, отступы, цвета, шрифты, состояния элементов) (15). • Макет предоставлен, но есть небольшие недочеты (например, отсутствие некоторых размеров или состояний элементов) (10). • Макет предоставлен, но недостаточно детализирован (например, только основные экраны без учета всех состояний) (5). • Макет отсутствует (0).	15
K12	Соответствие макетам	Выбрать одно: • Верстка полностью соответствует созданному макету. Все элементы расположены правильно, соблюдены размеры, отступы, цвета и шрифты (10). • Есть незначительные отклонения от макета (например, небольшие расхождения в отступах или размерах элементов) (5). • Верстка не соответствует макету Figma: элементы расположены неправильно, размеры, отступы или цвета не соблюдены (0).	10
K13	Объем самостоятельной работы	Оценка самостоятельной работы каждого участника	0-100%

Формула расчета итогового балла:

$$(K1 + K2 + \cdots + K11 + K12) \times K13.$$

Максимальный балл — 100.

4.3.7. Решение задачи

Описание решения для серверной части

Ссылка с исходным кодом: https://gitnto.innovationcampus.ru/Bugdroid/Bugdroid-Back.

Серверная часть реализована с использованием Spring Boot. Написанный микросервис обернут в Docker-контейнер. Все миграции БД осуществляются при помощи `liquibase`, что при необходимости позволяет легко перенести схему и данные из одной базы в другую. В качестве основной реляционной СУБД выбрана PostgreSQL. Серверная часть, реализованная backend-разработчиком, полностью удовлетворяет ТЗ финального этапа.

База данных

В разработанной схеме база данных (далее БД) все данные атомарны и исключены транзитивные зависимости. Корректно расставлены связи между таблицами.

Архитектура микросервиса

Код серверной части грамотно разделен на три слоя:

- Controller,
- Service,
- Repository.

Используются принципы SOLID.

Существует разделение на сущности (классы слоя работы с БД) и DTO (классы слоя общения с клиентской частью).

Описанное выше позволяет легко модифицировать и расширять полученное решение.

Обработка исключений

Реализована централизованная обработка ошибок, возникающих на стороне сервера.

Созданы кастомные исключения, что показывает осознанное принятие решения об обработке исключительных ситуаций.

Все исключения содержат точное описание, которое позволит на стороне клиентской части сформировать понятное пользователю сообщение об ошибке.

Аутентификация и авторизация

В качестве аутентификации и авторизации выбрана Basic Spring Security.

Данные о пользователе хранятся в базе в зашифрованном виде.

Реализовано разделение ролей (USER, ADMIN).

Роли вынесены в отдельную таблицу БД, что упрощает расширение ролевой модели в случае необходимости.

Оптимизация

Реализована пагинация, что позволяет осуществить постраничную выгрузку данных.

Тесты и документация

В проекте написаны интеграционные тесты, которые проверяют все три слоя (Controller, Service, Repository).

Присутствует Swagger-документация с расширенным описанием API.

Описание решения для клиентской части

Ссылка с исходным кодом: https://gitnto.innovationcampus.ru/Bugdroid/Bugdroid-Front.

Android приложение реализовано на языке Kotlin с использованием технологий:

- XML для создания верстки;
- RecyclerView для отрисовки списков;
- Paging 3 для реализации пагинации;
- Ktor client для реализации клиент-серверного взаимодействия.

Проект разделен на слои: `data`, `domain`, `ui (presentation)`, что позволяет масштабировать его в дальнейшем и соответствует современным стандартам индустрии в архитектуре Android-приложений.

Согласно техническому заданию, команда успешно выполнила:

- авторизацию с паролем и применила ее для всех запросов в приложении;
- постраничную загрузку списка с информацией о последних входах, в котором было обработано сохранение состояния позиции, а также наличие всей информации, согласно техническому заданию; элементы переиспользовались;
- экран администратора, на котором есть возможность осуществить поиск сотрудника по логину с возможностью блокировки пропуска.

Сильные стороны проекта

Интерфейс пользователя

Разработанное приложение полностью соответствует требованиям по наличию всех базовых элементов интерфейса. Экран авторизации предоставляет понятный интерфейс для входа с использованием логина и пароля. Главный экран реализует отображение списка проходов с использованием RecyclerView, а встроенный функционал сканирования QR-кода обеспечивает основной механизм идентификации пользователей. Все элементы интерфейса логично связаны между собой, имеют единообразный дизайн и обеспечивают интуитивно понятную навигацию.

Авторизация и безопасность

Команда реализовала механизм Basic-авторизации, который применяется ко всем сетевым запросам в приложении. Система авторизации также интегрирована с функционалом разграничения доступа, что позволяет корректно определять и ограничивать возможности обычных пользователей и администраторов.

Список проходов и работа с данными

Одной из сильнейших сторон проекта является эффективная реализация списка проходов с использованием технологии Paging 3. Этот подход обеспечивает оптимизацию сетевого трафика и памяти устройства благодаря загрузке только необходимых данных при прокрутке списка.

Команда позаботилась о сохранении позиции прокрутки при смене конфигурации устройства, что значительно улучшает пользовательский опыт. Все элементы списка содержат полную информацию в соответствии с требованиями (время сканирования, идентификатор считывателя, тип прохода) и эффективно переиспользуются благодаря правильной реализации адаптера RecyclerView.

Функциональность администратора

Экран администратора реализует все необходимые функции согласно техническому заданию. Интерфейс позволяет осуществлять поиск сотрудников по логину и предоставляет доступ к информации о каждом пользователе.

Особенно стоит отметить функциональность блокировки и разблокировки пропуска, которая не только изменяет статус в системе, но и корректно обрабатывается при последующих попытках сканирования QR-кода.

Реализована также защита от несанкционированного доступа — раздел администратора визуально скрыт для обычных пользователей, что повышает безопасность и улучшает пользовательский опыт.

Обработка ошибок и сетевых проблем

Приложение демонстрирует высокий уровень устойчивости к сбоям благодаря продуманной системе обработки ошибок. На экране авторизации выводятся сообщения в случае проблем с входом. Реализована корректная обработка отсутствия интернет-соединения как на главном экране.

Архитектура и код

В архитектуре приложения применен паттерн MVVM, что обеспечивает четкое разделение бизнес-логики и пользовательского интерфейса. Это решение существенно упрощает тестирование и дальнейшую поддержку кода.

Команда правильно выстроила зависимости между слоями, что минимизирует связанность компонентов. Код хорошо декомпозирован на классы и методы с четким разделением ответственности.

Весь проект написан с использованием современных библиотек и API, без применения устаревших решений, что обеспечивает производительность и совместимость с новыми версиями Android.

4.3.8. Материалы для подготовки

Буткемпы IT-школы

1. Буткемп IT школы Samsung 2023 г. [Электронный ресурс]: видеокурс / YouTube. — URL: https://www.youtube.com/playlist?list=PLa2T1zmZ6w5IEH3toqEA8wXPXzvNhkmQ2.
2. Буткемп IT-школы Samsung 2024 г. [Электронный ресурс]: видеокурс / YouTube. — URL: https://www.youtube.com/playlist?list=PLa2T1zmZ6w5LvGpFM9whAqkHB9Iq38cSn.

Курс по Java

3. Курс по Java [Электронный ресурс]: обучающие материалы / Яндекс.Диск.

4. Модуль 1. — URL: https://disk.yandex.ru/i/w6y7vcEfpygzZg.
5. Модуль 2. — URL: https://disk.yandex.ru/i/TR4Ubr1HWkqqMw.
6. Модуль 3. — URL: https://disk.yandex.ru/i/n1U7ISb6EzZ6_w.
7. Модуль 4. — URL: https://disk.yandex.ru/i/CbXW6VygzLqeaQ.
8. Модуль 5. — URL: https://disk.yandex.ru/i/X9nuXVJG5eLakw.
9. Модуль 6. — URL: https://disk.yandex.ru/i/WP3gV5nBb4ksvg.
10. Модуль 7. — URL: https://disk.yandex.ru/i/OIO-L5ZPQ74_Rg.
11. Модуль 8. — URL: https://disk.yandex.ru/i/nWrw9nYtdX-Crg.
12. Модуль 9. — Ч. 1: https://disk.yandex.ru/i/01C71e2PKx2Htw; Ч. 2: https://disk.yandex.ru/i/mQHYwTy12aHC-A.
13. Модуль 10. — URL: https://disk.yandex.ru/i/RfmOYylnVr3RhQ.
14. Модуль 11. — URL: https://disk.yandex.ru/i/Pg0bhPANFsjtKQ.

Курс по Spring Framework

15. Курс по Spring Framework [Электронный ресурс]: обучающие материалы / Яндекс.Диск.
16. Модуль 1. — Ч. 1: https://disk.yandex.ru/i/q4Df2mH-0KI2Ug; Ч. 2: https://disk.yandex.ru/i/gb2RK9L7BOC89w.
17. Модуль 2. — Ч. 1: https://disk.yandex.ru/i/E34JfcAK8nA4Vw; Ч. 2: https://disk.yandex.ru/i/9TNQ8D5V3cPjrg.
18. Модуль 3. — Ч. 1: https://disk.yandex.ru/i/-pBq185elh1a9Q; Ч. 2: https://disk.yandex.ru/i/3WpYIOxD-26GiQ.
19. Модуль 4. — Ч. 1: https://disk.yandex.ru/i/Yn5EWPLotD7EPg; Ч. 2: https://disk.yandex.ru/i/wJR4SB2fVxMhLA; Ч. 3: https://disk.yandex.ru/i/UgHtBE-hS64Img.
20. Модуль 5. — Ч. 1: https://disk.yandex.ru/i/YnjoEBPRF9htHQ; Ч. 2: https://disk.yandex.ru/i/5_V2B5qIimFyJg.
21. Модуль 6. — URL: https://disk.yandex.ru/i/hM_Jhiw2OZw0CA.
22. Модуль 7. — Ч. 1: https://disk.yandex.ru/i/840Kmro-SBRK_Q; Ч. 2: https://disk.yandex.ru/i/_ASfB9ZxtMnADA; Ч. 3: https://disk.yandex.ru/i/Q0UXf700ycN8zQ.

Курсы по Kotlin

23. Kotlin Bootcamp for programmers [Электронный ресурс]: онлайн-курс / Android Developers. — URL: https://developer.android.com/courses/kotlin-bootcamp/overview.
24. Developing Android Apps with Kotlin [Электронный ресурс]: онлайн-курс / Udacity. — URL: https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012.

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 туру не менее 30,0 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 33,0 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 100,00 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 $K_1 = 0{,}1$, по

предмету 2 (информатика) с весом $K_2 = 0{,}2$, 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 3 победителя и 7 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	62,30 и выше
Призеры	От 52,05 до 62,10

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 (P2P). 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».