



НТО

МАТЕРИАЛЫ ЗАДАНИЙ
Всероссийской междисциплинарной олимпиады
школьников 8–11 класса
«Национальная технологическая олимпиада»
по профилю
«Технологии беспроводной связи»

2024/25 учебный год

ntcontest.ru

УДК 373.5.016:621.391
ББК 74.263.2
Т38

Авторы:

И. А. Воронцов, А. А. Гаврилюк, Е. Н. Горечин, И. С. Давыдов, О. В. Зубков,
О. Д. Иванова, К. Д. Кириченко, М. Ю. Просекин, И. Г. Просекина, А. Э. Ржечицкий,
Д. М. Цивилева, М. А. Чекан, В. В. Широков

Т38 Всероссийская междисциплинарная олимпиада школьников 8–11 класса
«Национальная технологическая олимпиада». Учебно-методическое пособие
Том 19 **Технологии беспроводной связи**
— М.: Ассоциация участников технологических кружков, 2025. — 185 с.

ISBN 978-5-908021-18-0

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:

02.00.00 Компьютерные и информационные науки

09.00.00 Информатика и вычислительная техника

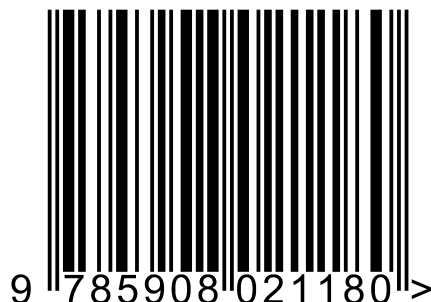
10.00.00 Информационная безопасность

11.00.00 Электроника, радиотехника и системы связи

27.00.00 Управление в технических системах

ISBN 978-5-908021-18-0

УДК 373.5.016:621.391
ББК 74.263.2



9 785908 021180 >

Оглавление

1 Введение	5
1.1 Национальная технологическая олимпиада	5
1.2 Технологии беспроводной связи	13
2 Первый отборочный этап	17
2.1 Работа наставника НТО на этапе	17
2.2 Предметный тур. Информатика	18
2.2.1 Первая волна. Задачи 8–11 класса	18
2.2.2 Вторая волна. Задачи 8–11 класса	28
2.2.3 Третья волна. Задачи 8–11 класса	38
2.2.4 Четвертая волна. Задачи 8–11 класса	51
2.3 Предметный тур. Математика	66
2.3.1 Первая волна. Задачи 8–9 класса	66
2.3.2 Первая волна. Задачи 10–11 класса	69
2.3.3 Вторая волна. Задачи 8–9 класса	73
2.3.4 Вторая волна. Задачи 10–11 класса	76
2.3.5 Третья волна. Задачи 8–9 класса	81
2.3.6 Третья волна. Задачи 10–11 класса	86
2.3.7 Четвертая волна. Задачи 8–9 класса	90
2.3.8 Четвертая волна. Задачи 10–11 класса	94
2.4 Инженерный тур	99
3 Второй отборочный этап	107
3.1 Работа наставника НТО на этапе	107
3.2 Инженерный тур	109
3.2.1 Командные задачи	112

4	Заключительный этап	125
4.1	Работа наставника НТО при подготовке к этапу	125
4.2	Предметный тур	127
4.2.1	Информатика. 8–11 классы	127
4.2.2	Математика. 8–9 классы	142
4.2.3	Математика. 10–11 классы	146
4.3	Инженерный тур	153
4.3.1	Общая информация	153
4.3.2	Легенда задачи	153
4.3.3	Требования к команде и компетенциям участников	154
4.3.4	Оборудование и программное обеспечение	154
4.3.5	Описание задачи	158
4.3.6	Материалы для подготовки	181
5	Критерии определения победителей и призеров	183
6	Работа наставника после НТО	185

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. Технологии беспроводной связи

Профиль Технологии беспроводной связи напрямую связан с потребностью современного мира в надежной и производительной связи в условиях постоянного роста числа подключаемых устройств и объемов передаваемого трафика. Ключевыми перспективными областями отрасли являются космическая сфера, промышленность, подводная и мобильная робототехника.

Чем дальше развиваются технологии, тем сильнее люди оказываются в ситуации, когда, имея большой пользовательский опыт, они не понимают, как работают эти технологии. Поэтому вопросы цифровой грамотности, безопасности данных, безопасности и развития инфраструктур становятся все более актуальными.

Задачи инженерного (практического) тура профиля перекликаются с актуальными задачами систем связи: от работы с различными форматами данных и организации помехоустойчивого кодирования для передачи информации в условиях шумов до разработки адаптивных систем слежения и протоколов связи.

Команды работают со слежением за подвижным объектом и источником сигнала, кодированием и декодированием сигналов, изучают характер шумов, а также создают собственные протоколы передачи данных с учетом определенных требований к надежности, характеру помех в канале, скорости передачи и взаимосвязи данных в многоканальных системах.

Главный вызов настоящего времени — технологический. Критически необходимым является развитие сквозных технологий, а также инфраструктурное обеспечение технологического развития. Это создает условия принципиально нового качества устойчивого экономического роста за пределами периода восстановления.

К числу сквозных технологий относятся технологии беспроводной связи, касающиеся как мобильных сетей передачи данных 5G, так и каналов связи спутников и интернета вещей IoT. Именно этому направлению посвящен профиль. Внедрение цифровых технологий и развитие отечественных ИТ-решений в отрасли беспроводной связи является частью цифровой трансформации — одной из национальных целей развития Российской Федерации.

Знакомство школьников с современной инженерной задачей во время соревнований и подготовки к ним позволяет не просто ввести их в курс дела, но дает возможность получить личный практический опыт решения небольших, но реальных задач в данной области.

Навык создания собственных протоколов — базовый в технологиях связи, огромном растущем рынке в мире с увеличивающимся спросом на специалистов. Дальнейшее развитие беспроводных технологий связи — это не только технологии связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов и других объектов интернета вещей — все они требуют создания протоколов связи.

В настоящее время количество различных протоколов связи, а также систем, где их важно проектировать, непрерывно растет, но понимание базовых принципов и способность конструировать такие каналы падает в связи с ростом сложности протоколов. Поэтому важно работать, с одной стороны, с максимально современной

постановкой задачи, а с другой — с максимальной прозрачностью и возможностью ее увидеть и «пощупать».

Знакомство с профилем начинается с «Урока НТО» по профилю Технологии беспроводной связи, который проводится в средних общеобразовательных учреждениях. Материалы для его проведения размещены на сайте <https://nto-lesson.ru/>.

Задачи профиля составлены таким образом, что для их решения требуются не только уровень школьной подготовки, но и углубленные знания в области программирования, математики и геометрии, а также основные представления в области помехоустойчивого кодирования.

От этапа к этапу увеличивается сложность и специфика этих задач. По мере продвижения к заключительному этапу проводятся вебинары, предоставляются дополнительные методические материалы по сложным темам:

- методы исследования каналов связи и обработки сигналов;
- методы борьбы с шумами;
- получение практических навыков по помехоустойчивому кодированию в системах связи;
- практика работы с двоичными данными на уровне байтов и битов;
- практика анализа текстовой и графической информации.

Для их освоения созданы онлайн-курсы от разработчиков по тематике профиля.

Первый отборочный этап (дистанционный индивидуальный) состоит из предметного и инженерного туров.

Предметный тур определяет уровень подготовки школьников по математике и информатике, устанавливая необходимую планку для участия во втором этапе и работы с обучающими материалами.

Задачи инженерного тура первого этапа сформулированы таким образом, чтобы отразить специфику профиля. Они проще, чем задачи второго тура, и затрагивают актуальные темы в системах связи, но (в отличие от второго тура) могут быть решены участниками индивидуально.

Второй этап — это не только отборочное соревнование, но и введение в профиль. Здесь задачи сформулированы таким образом, чтобы отразить специфику профиля, они затрагивают актуальные темы в системах связи, а также готовят команды к финальным испытаниям и способствуют развитию элементов научного исследовательского подхода (работа со сложными моделями, информационный поиск и нестандартное мышление) и навыков командной работы.

Все задачи этапа сочетают в себе математику и информатику. Для их решения необходима не только школьная подготовка, но и факультативные знания в области программирования, математики и геометрии — методические рекомендации позволяют очертить область знаний и навыков для самостоятельного изучения. Второй этап полностью построен вокруг компетенций и ролей, которые требуются на заключительном этапе.

Совместная деятельность участников начинается во время **командного этапа второго тура**. Помимо общих механизмов НТО для профиля создано условие, направленное на командное взаимодействие, — ограниченное число попыток решения задач. Это приводит к необходимости уже на начальном этапе договариваться об общекомандной стратегии сдачи решений. Второй тур предлагается завершать ко-

мандам совместной рефлексией и выявлением слабых и сильных сторон у каждого члена команды, а также белых пятен в знаниях и навыках.

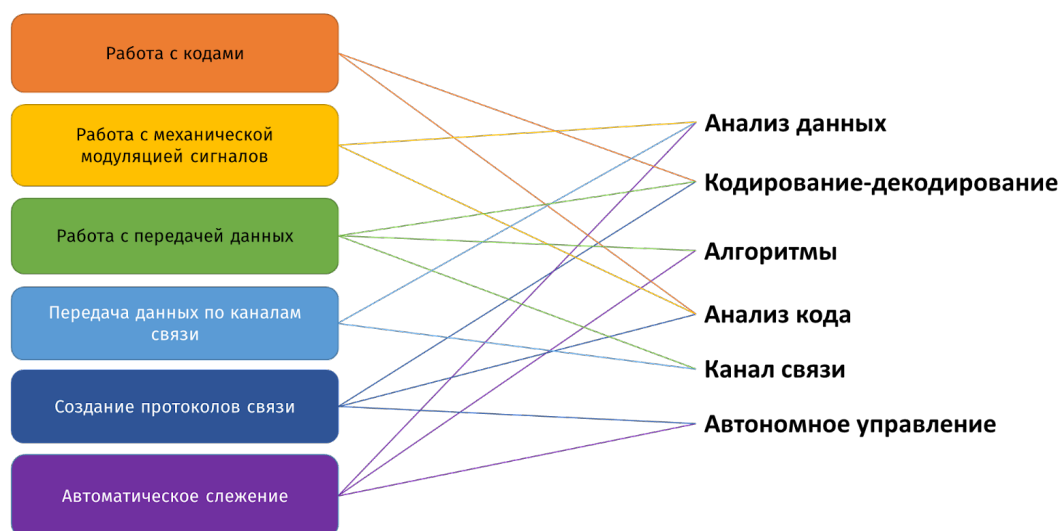


Рис. 1.2.1. Взаимосвязь задач инженерного командного тура и тем задач второго этапа

Заключительный этап профиля состоит из семи задач. Задачи командной части, с одной стороны, независимы, а другой — взаимосвязаны между собой и с финальной задачей. Количество одновременно решаемых задач определяет сложность командной работы.

Вся работа проводится в рамках регламента, ограничивающего временные слоты, это побуждает участников активизировать совместную деятельность, планировать командное распределение задач с самого начала заключительного этапа, а кроме того, вносит игровой соревновательный элемент.

Заключительный этап 2024–25 гг. проходит на комплексе стендов, взаимосвязанных по компетенциям и моделирующих реальную задачу передачи данных по физическому каналу, анализа данных и создания помехозащищенных протоколов. Итог работы — набор протоколов передачи данных с учетом физических особенностей среды.

На инженерном туре команды профиля Технологии беспроводной связи работают со стендами:

- «Узконаправленные низкоэнергетические каналы связи» — УНКС. Программно-аппаратная платформа стенда УНКС позволяет работать с адаптивными алгоритмами и точностью передачи сигнала. При создании стенда разработчиками и экспертами выступали специалисты ИСЗФ СО РАН с реальным опытом работы проектирования систем слежения — современных радаров.
- «Каналы связи и кодирование» — О-БТС. Стенды УНКС и О-БТС входят в образовательный комплекс «Беспроводные технологии связи», разработанный компанией ИнСитиЛаб, и созданы в логике моделирования реальных инженерных задач по созданию протоколов связи. Стенд О-БТС стал нововведением, добавив задачи на оптический канал связи.
- «Турнир Юных Киберфизиков «Акустика» — ТЮК-А. На комплекте для проведения турнира юных киберфизиков «Акустика-1», разработанном компанией

«ИнСитиЛаб» в рамках Национальной киберфизической платформы, участники взаимодействуют с акустическим каналом связи. Комплект ТЮК-А позволяет наглядно работать с аналоговыми и цифровыми сигналами в физическом акустическом канале связи, с разными типами модуляции и помехоустойчивым кодированием.

- «Акустический полигон» — А-Полигон. Еще одно нововведение, модификация ТЮК-А с шумоизолированным боксом. Стенд позволяет ставить комплексные задачи на акустическом канале, на которые не повлияет окружающий шум.

Участникам ставится задача в виде описаний моделей, интерфейсов взаимодействия с ними и критерия оценки успешности решения отдельных задач. Командам необходимо выполнить серию заданий в указанном формате, который используется для проведения измерения, формирующего результат решения. Ограничения на решения обусловлены моделью, техническими параметрами стендов. Итог — успешно декодированное сообщение; набор программ, реализующих протокол или слежение, и выполняющих поставленную задачу с достаточной точностью.

Помимо командной задачи (практический тур), на заключительном этапе участники решают индивидуальные задачи по предметам: информатика и математика (предметный тур). Здесь преследуются следующие цели:

- объективно проверить индивидуальные знания участников;
- косвенно оценить индивидуальный вклад участников в результат командной работы.

Удельный вес этих задач в финальной оценке участников составляет 40%.

Таким образом, цепочка туров профиля позволяет выявить как сильных участников, так и соблюсти основной принцип, заложенный в основу второго и заключительного этапов профиля, — необходимость командной работы.

Уровень заданий, их взаимосвязь и организационные регламенты направлены на формирование команд, способных решать новые инженерные задачи за ограниченное время, каждый член которых имеет хорошие знания и навыки в профильных предметах.

Победители и призеры профиля имеют возможность поступить в Московский технический университет связи и информатики (МТУСИ) без вступительных испытаний.

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 $0 \leq t \leq 10^6$.

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
9999970000002000000

Решение

Ниже представлено решение на языке C++.

C++

```

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

Задача 2.2.1.2. Двойное остекление (15 баллов)

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

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++

```

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

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

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Формат входных данных

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

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

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

Формат выходных данных

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

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

Примеры

Пример №1

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

Комментарий

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

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

```

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

```

Задача 2.2.1.4. Бонусы и экономия (25 баллов)

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

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

Формат входных данных

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

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++

```

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

```

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

```

Задача 2.2.1.5. Сон таксиста (30 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

В связи с такими странными правилами, таксистам в этом городе разрешено законом везти пассажира по любому маршруту, не нарушающему направления движения. Все в этом городе привыкли к такой ситуации и абсолютно спокойно относятся к тому, что таксисты везут их самым длинным путем. Разумеется, 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 $2 \leq n, m \leq 2 \cdot 10^5$.

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

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

Формат выходных данных

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++

```

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

```

```

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

```

2.2.2. Вторая волна. Задачи 8–11 класса

Задачи второй волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контеcт: <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 $n \times m$.

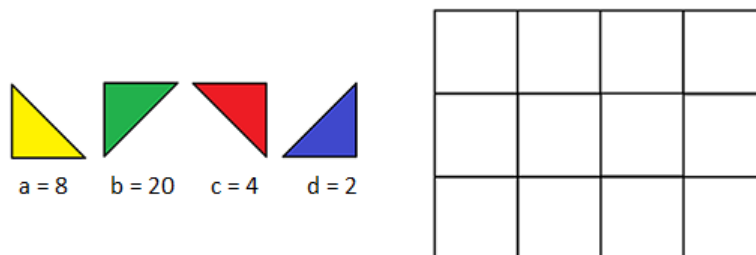


Рис. 2.2.1

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.

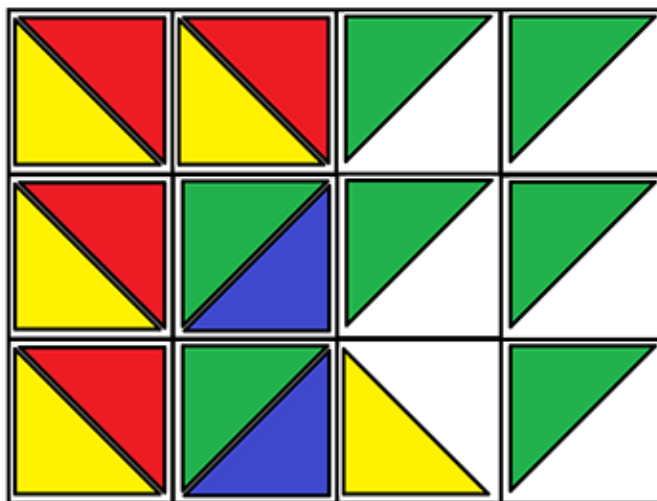


Рис. 2.2.2

Решение

Ниже представлено решение на языке C++.

C++

```

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

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

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одна старая задача имеет следующий вид:

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

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

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++

```

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

Задача 2.2.2.3. Ладья и обязательная клетка (20 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Стандартный ввод
3 4 2 3
Стандартный вывод
6

Примечания

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

Комментарий

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

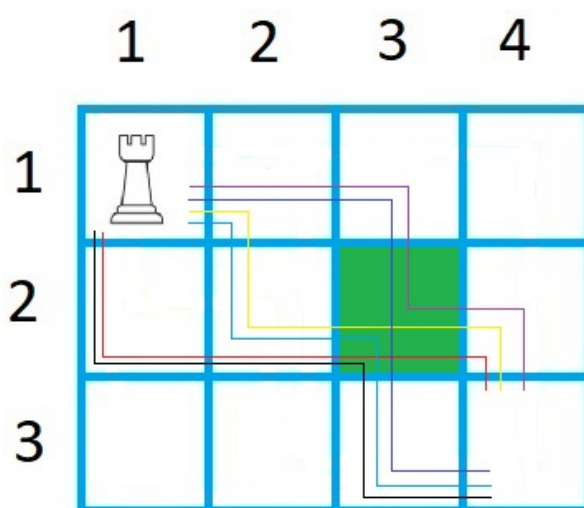


Рис. 2.2.3

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

```

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

```

Задача 2.2.2.4. Танец с цифрами (25 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Десять танцоров репетируют на сцене новый танец. Каждый танцор одет в 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 появиться 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++

```

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

```

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

```

Задача 2.2.2.5. Трудная сортировка (30 баллов)

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

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

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

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

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

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

Формат входных данных

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

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

Формат выходных данных

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

Примеры

Пример №1

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++

```

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

2.2.3. Третья волна. Задачи 8–11 класса

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

Задача 2.2.3.1. Туннель (10 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

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

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

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

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

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

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

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

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

считаем, что работает первая стратегия.

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

Стандартный ввод
1 5 10
Стандартный вывод
3

Решение

Ниже представлено решение на языке C++.

C++

```

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

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

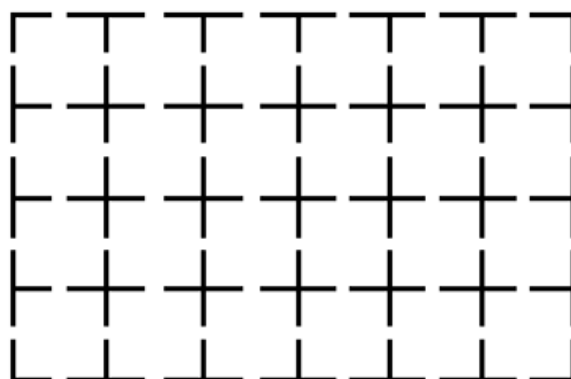


Рис. 2.2.4

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

Стандартный ввод
16 15
Стандартный вывод
4 6

Пример №2

Стандартный ввод
0 0
Стандартный вывод
1 1

Комментарий

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

Решение

Ниже представлено решение на языке C++ при помощи бинарного поиска.

C++

```

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

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

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

«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 $2 \leq n, m \leq 1000$.

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

Формат выходных данных

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

Примеры

Пример №1

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.

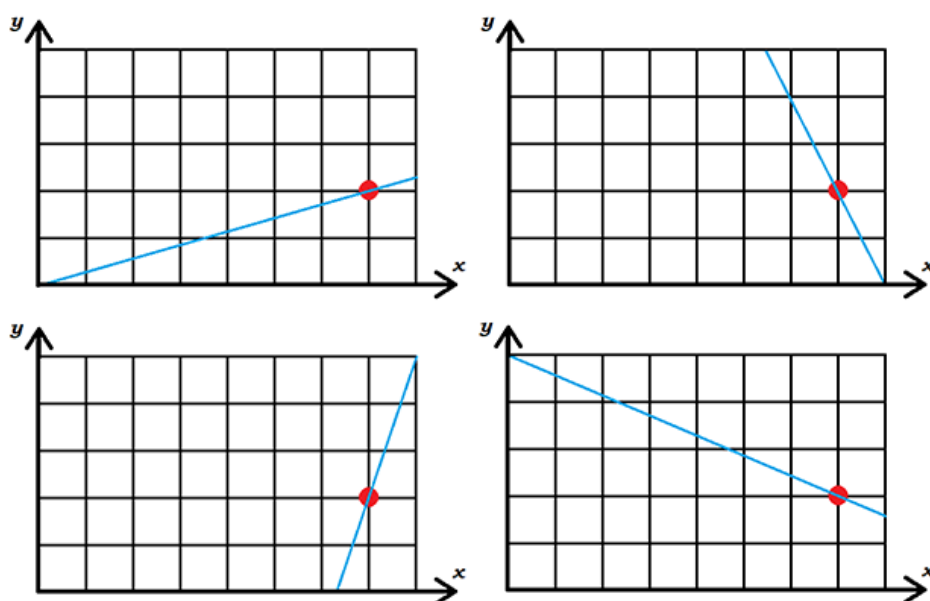


Рис. 2.2.5

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++

```

1  #include<bits/stdc++.h>
2  #define int long long
3  using namespace std;
4  const int INF = 1e18;
5  double katy(double x, double y, double n){
6      return n * y / x;
7  }
8  double n, m, x, y;
9  double ans = INF;
10 double k1, k2;
11 void upd(){
12     if(k1 < m){
13         double st = k1 * n / 2;
14         ans = min(ans, n * m - 2 * st);
15     }
16     else{
17         double st = k2 * m / 2;
18         ans = min(ans, n * m - 2 * st);
19     }
20 }
21 signed main(){
22     cin >> n >> m >> x >> y;
23     k1 = katy(x, y, n);
24     k2 = katy(y, x, m);
25     upd();
26     k1 = katy(n - x, y, n);
27     k2 = katy(y, n - x, m);
28     upd();
29     k1 = katy(x, m - y, n);
30     k2 = katy(m - y, x, m);
31     upd();
32     k1 = katy(n - x, m - y, n);
33     k2 = katy(m - y, n - x, m);
34     upd();
35     cout.precision(3);
36     cout << fixed << ans<< endl;
37 }
```

Задача 2.2.3.4. Плетенка (25 баллов)

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

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.

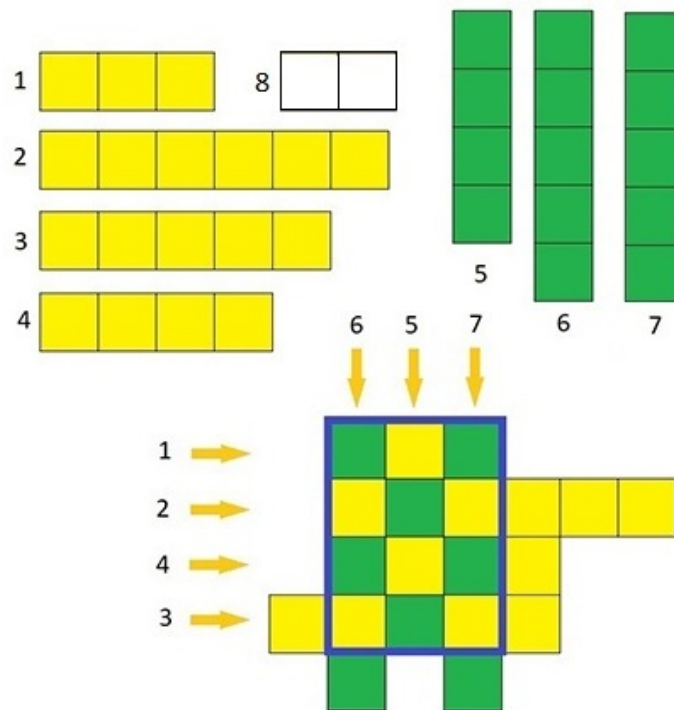


Рис. 2.2.6

Решение

Ниже представлено решение на языке C++.

C++

```

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

Задача 2.2.3.5. Английский в игровой форме (30 баллов)

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

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

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Формат входных данных

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

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

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

Формат выходных данных

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

Примеры**Пример №1**

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

Комментарий

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

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

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

2.2.4. Четвертая волна. Задачи 8–11 класса

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

Задача 2.2.4.1. Квадратный флаг (10 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

Формат входных данных

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

Формат выходных данных

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

Примеры*Пример №1*

Стандартный ввод
2 4
3 6
Стандартный вывод
4

Пример №2

Стандартный ввод
2 2
3 6
Стандартный вывод
3

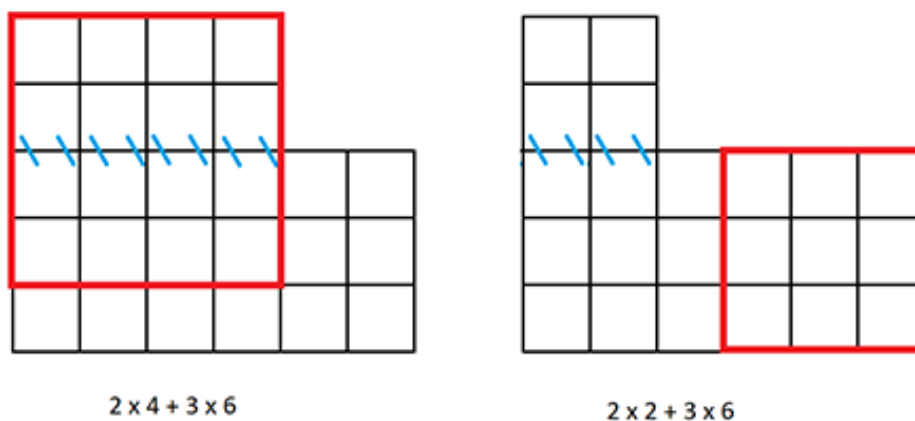
Примечания

Рис. 2.2.7

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

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

Задача 2.2.4.2. Потерянная ДНК (15 баллов)

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Например, пусть $D_1 = \mathbf{AACGGTAGGT}$, а $D_2 = \mathbf{TCCCGGAACA}$. Выберем $i = 4$ и поменяем местами префиксы длины 4. Получим две новые ДНК, одна из которых будет иметь вид $\mathbf{AACGGGAACA}$, а вторая — $\mathbf{TCCCGTAGGT}$. 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 $10 \leq n \leq 100$, $2 \leq m \leq 100$.

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

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

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

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

Формат выходных данных

Вывести недостающую утерянную ДНК.

Примеры

Пример №1

Стандартный ввод
10 2
AACGGTAGGT
TCCCGGAACA

TCCCGTAGGT
Стандартный вывод
AACGGGAACA

Пример №2

Стандартный ввод
10 4
AACCGGTAA
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++

```

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

Задача 2.2.4.3. Утомленные туристы (20 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

У каждого участника своя скорость движения в туннеле. Пусть участники проходят туннель за A , B , C и D мин. Если идут двое, то они движутся со скоростью того, кто идет медленнее. Требуется по заданным 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 $1 \leq A, B, C, D \leq 1000$, $0 \leq E \leq 1000$.

Формат выходных данных

Вывести одно число — минимальное время прохождения туннеля всей группой.

Примеры

Пример №1

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++

```

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

```

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

```

Задача 2.2.4.4. Проектируем мост (25 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

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

Формат входных данных

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

Формат выходных данных

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

Примеры

Пример №1

Стандартный ввод
4 3
Стандартный вывод
7

Примечания

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

```

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

```

Комментарий

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

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

```

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

```

Задача 2.2.4.5. Джентльмены на прогулке (30 баллов)

Имя входного файла: стандартный ввод или input.txt.

Имя выходного файла: стандартный вывод или output.txt.

Ограничение по времени выполнения программы: 8 с.

Ограничение по памяти: 64 Мбайт.

Условие

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

При каждой встрече два джентльмена приветствуют друг друга, приподнимая

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

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

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

Формат входных данных

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

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

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

Формат выходных данных

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

Примеры

Пример №1

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

Пример №2

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.

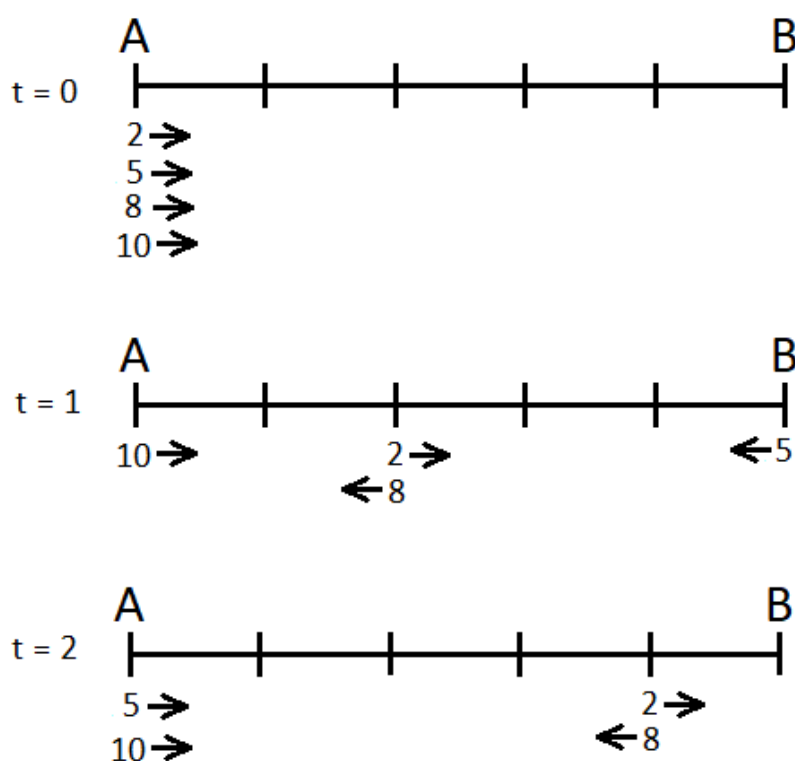


Рис. 2.2.8

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 бинарного поиска с квадратичным нахождением ответа в каждой его итерации.

Решение

Ниже представлено решение на языке C++.

C++

```

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

```

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

```

2.3. Предметный тур. Математика

2.3.1. Первая волна. Задачи 8–9 класса

Задачи первой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contests.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

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

Тогда

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

Ответ: 0,25.

Задача 2.3.1.5. (25 баллов)

Тема: теория вероятностей.

Условие

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

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

Решение

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

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

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

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

Ответ: 0,0002.

2.3.2. Первая волна. Задачи 10–11 класса

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

Задача 2.3.2.1. (10 баллов)

Темы: комбинаторика, десятичная запись числа, цифры.

Условие

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 съесть $60 \cdot 0,35 = 21$ 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 получают 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

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

Тогда, подставив их в уравнение первой параболы, получаем:

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

Отсюда $a = -1$.

Ответ: -10 .

Задача 2.3.2.4. (25 баллов)

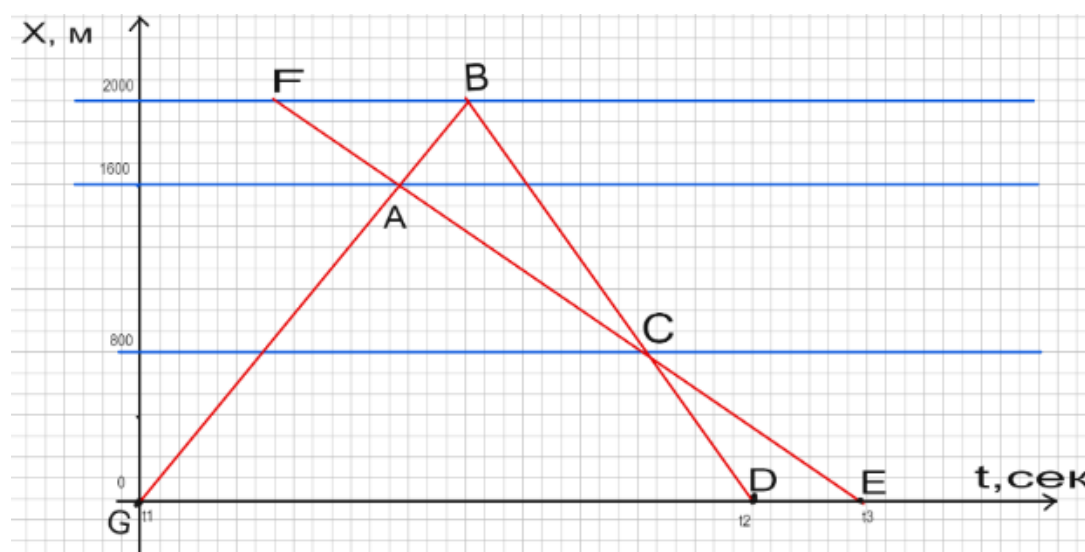
Темы: текстовые задачи и логика, графическое изображение движения, теорема Менелая.

Условие

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

Решение

Самое изящное решение получается графическим методом.



Здесь данные из условия задачи можно обозначить следующим образом:

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

По теореме Менелая получаем:

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

Выразим эти отношения из разных пар подобных треугольников:

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

Подставим числа:

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

Ответ: 12.

Задача 2.3.2.5. (30 баллов)

Условие

Инженер-исследователь работает над созданием новой системы гиперпространственной навигации для космических кораблей, которая потребует меньших вычислительных ресурсов. Часть измерений гиперпространства скрыта от нас и устроена не так, как мы привыкли, а именно — являются дискретными (с конечным количеством позиций), 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 $0 - 1 - 2 - 3 - 4 - 0 - \dots$ (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 выбрать 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.Контеcт: <https://contests.yandex.ru/contest/63460/enter/>.

Задача 2.3.3.1. (15 баллов)

Тема: арифметика.

Условие

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

Во сколько раз скорость первого поезда больше скорости второго?

Решение

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

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

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

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

Ответ: 4.

Задача 2.3.3.2. (15 баллов)

Тема: комбинаторика.

Условие

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

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

Решение

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

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

Ответ: 3905.

Задача 2.3.3.3. (20 баллов)

Тема: геометрия.

Условие

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

Решение

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

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

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

Ответ: 90° .

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

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

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

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

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

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

Ответ: 385.

Задача 2.3.3.5. (25 баллов)

Тема: текстовая задача.

Условие

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

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

Какое количество матчей «Математики» проиграли?

Решение

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

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

Найдем количество игр, которые были проиграны, или отменены:

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

Тогда количество игр, в которых были поражения, равно

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

Получили диофантово уравнение

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

Выразим x :

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

Таким образом, $y \div 11$ и $y > 0$.

Рассмотрим различные случаи относительно y :

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

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

Ответ: 11.

2.3.4. Вторая волна. Задачи 10–11 класса

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

Задача 2.3.4.1. (10 баллов)*Темы: стереометрия, центральная симметрия.***Условие**

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

Чему равно максимальное значение объема одной из этих двух частей?

Решение

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

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

Ответ: 15.

Задача 2.3.4.2. (15 баллов)*Темы: теорема Виета, неравенство о средних, многочлены.***Условие**

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

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

Здесь буквой a обозначен неизвестный коэффициент.

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

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

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

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

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

Решение

Запишем теорему Виета для квадратного уравнения:

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

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

Ответ: -10 .

Задача 2.3.4.3. (20 баллов)

Темы: арифметическая задача, симметрия.

Условие

Исследователи выращивают экспериментальную культуру грибов. Эти грибы размножаются почкованием. Гриб порождает два новых гриба каждые 4 ч. Только что появившийся гриб слишком маленький, и поэтому он должен еще 6 ч расти, прежде чем размножиться, 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

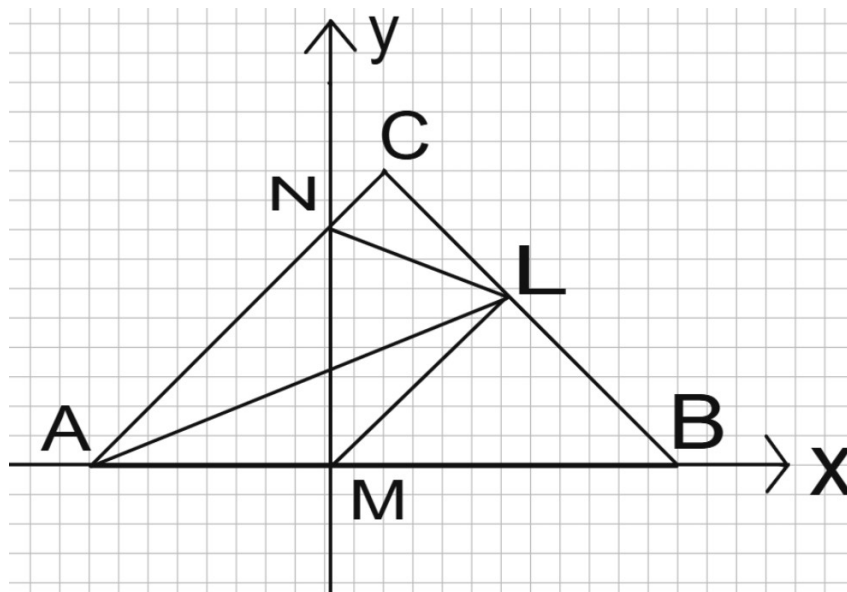
$MN = AM$, значит, угол ANM тоже равен 45° и NM перпендикулярно AB .

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:

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

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

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

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

Уравнение прямой, проходящей через эту точку перпендикулярно касательной:

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

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

Ответ: $(0; 0,015625)$.

2.3.5. Третья волна. Задачи 8–9 класса

Задачи третьей волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.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

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

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

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

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

Тогда чистая прибыль предпринимателя Петрова будет равна

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

Ответ: 691 200.

Задача 2.3.5.2. (15 баллов)*Тема: 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:

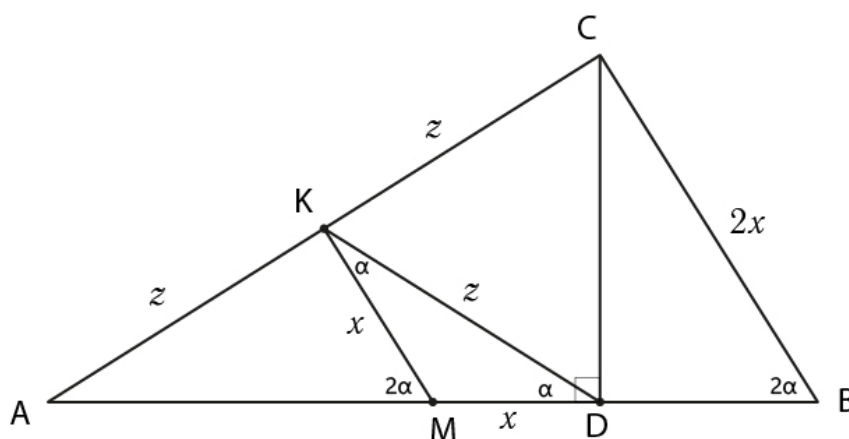
$$71 = x(n - 2k + m).$$

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

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

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

По теореме о внешнем угле для треугольника MKD

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

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

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

Ответ: 2.

2.3.6. Третья волна. Задачи 10–11 класса

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

Задача 2.3.6.1. (10 баллов)

Темы: осевая симметрия, равнобедренный треугольник, движение.

Условие

Известно, что выпуклая фигура Φ 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 Φ , равно дроби, в числителе которой шесть, а в знаменателе квадратный корень из числа π .

Чему равна площадь фигуры Φ ?

Решение

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

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

Следовательно, Φ является кругом радиуса 3, деленного на корень из числа π . Его площадь равна 9.

Ответ: 9.

Задача 2.3.6.2. (15 баллов)

Темы: составление уравнений, составление пропорций, проценты.

Условие

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

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

Решение

Найдем массу анобтаниума, оставшегося к концу первого часа.

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

Это составляет $100 - 80 = 20\%$.

Составим пропорцию и решим ее:

$$\begin{array}{l} 20\% - 900, \\ 100\% - ? \end{array}$$

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:

$$\begin{array}{l} 60\% - 4\,800, \\ 100\% - ? \end{array}$$

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)

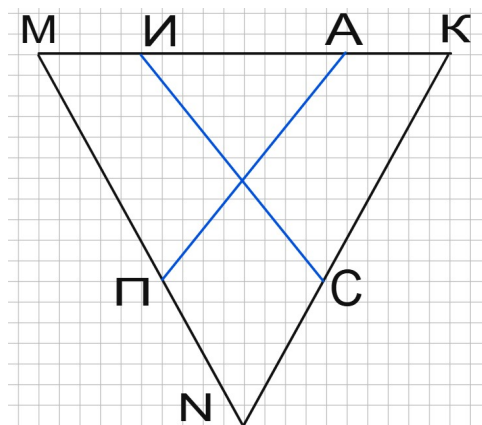
Темы: равносторонний треугольник, первый признак равенства треугольников.

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:

$MA + MP = KI + KC$, а значит, $MI + MP = KA + KC = 1\,000$ м.

$MI + KA = IA = 500$ м. Кроме того, длина KM равна 1 000 м.

Отсюда выходит, что $KA = 1\,000 - KC = 1\,000 - MA$, а значит, $KC = MA$. Аналогично выходит, что $KI = MP$.

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

Следовательно, $AP = SI$, а значит, $AP - SI + 0,5 \cdot IA = 250$ 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 $7 \cdot 1440 = 10\,080 > 10\,000$.

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.Контеcт: <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

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

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

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

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

Ответ: $-1,5$.

2.3.8. Четвертая волна. Задачи 10–11 класса

Задачи четвертой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контеcт: <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.

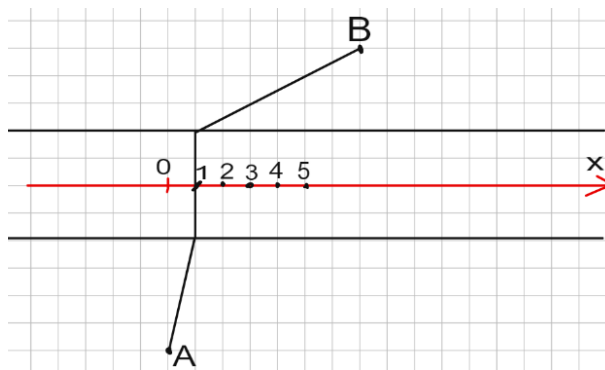


Рис. 2.3.1

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.

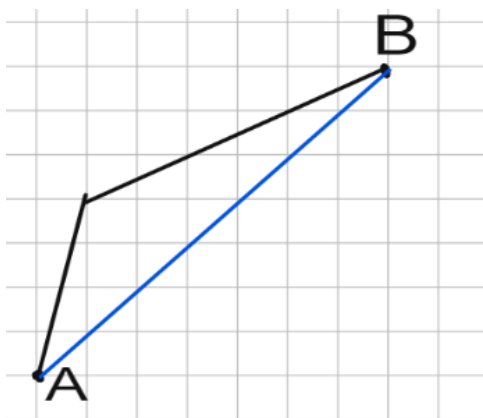


Рис. 2.3.2

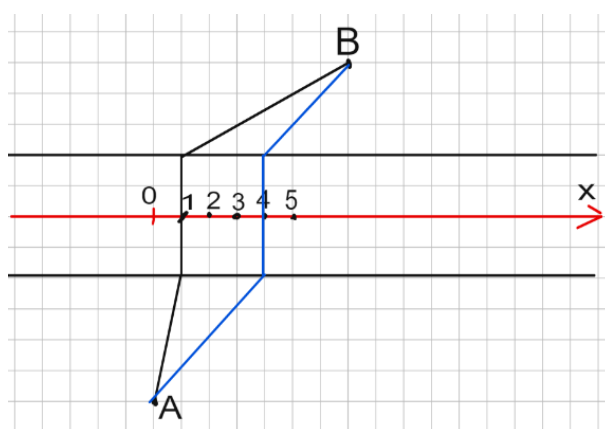


Рис. 2.3.3

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.

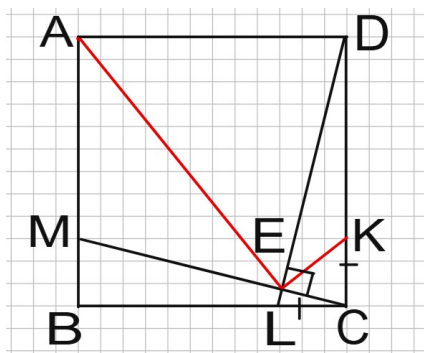


Рис. 2.3.4

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

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

Задачи первого этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контеcт: <https://contest.yandex.ru/contest/66698/enter/>.

Задача 2.4.1. Вкусный счет (10 баллов)

Тема: системы счисления.

Условие

Решите задачу на рис. 2.4.1 и запишите в ответ, чему равен X — число в десятичной системе счисления.

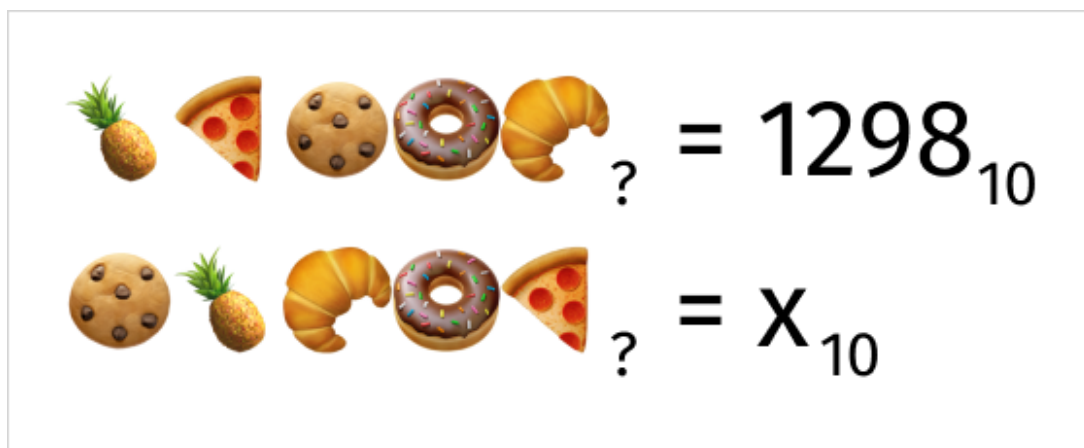


Рис. 2.4.1. Вкусный счет

Пояснение: все символы записаны в позиционной системе счисления, и она одинакова для верхнего и нижнего числа. Другие символы не предусмотрены.

Эта задача проверяет навык работы с математическими моделями и кодировками, что является частью задачи заключительного этапа.

Решение

В верхней строчке, слева от знака равенства, пять уникальных символов. Это означает, что сообщение создано в системе счисления с основанием 5 (по числу уникальных символов). При переводе 1298_{10} в систему счисления с основанием 5 получаем 20143_5 .

Сопоставляем символы с получившимся числом: «ананас» — 2, «пицца» — 0, «печенье» — 1, «пончик» — 4, «круассан» — 3. Из этого следует, что в нижней строчке слева от знака равенства стоит число 12340_5 .

Чтобы получить X , необходимо перевести данное число из системы счисления с основанием 5 в десятичную: $12340_5 = 970_{10}$. Записываем 970 в ответ.

Ответ: 970.

Задача 2.4.2. Ливерпульское кодирование (10 баллов)

Тема: каналы связи.

Условие

По восьми каналам связи закодировали и передали двоичные сообщения (рис. 2.4.2). Для формирования сигнала использовался принцип, очень похожий на манчестерское кодирование. Ниже приведены исходные данные до кодирования, которые передавали в канал, и принятые после прохождения каждого канала сигналы. При передаче по каналу данные могли исказиться, то есть передаться с ошибками. В ответ введите номер канала (только номер, без других символов) с наименьшим искажением данных.

Исходные данные	Полученные данные
1. 0000011011010011	
2. 1010100011010100	
3. 1010011011010010	
4. 1011001001010110	
5. 1011001011010100	
6. 1011010011011001	
7. 1011010011010001	
8. 1011100011010001	

Рис. 2.4.2. Ливерпульское кодирование

Эта задача проверяет навык анализа кодов и каналов связи, что является частью задачи заключительного этапа.

Решение

Манчестерское кодирование представляет собой кодировку двоичного разряда двумя тактами, где значение (ноль или единица) определяется характером изменения уровня между двух тактов. В системе, представленной в задаче, единица кодируется изменением уровня между двух тактов, а ноль — отсутствием изменения. По сути, это дифференциальное манчестерское кодирование со сдвигом на один такт вперед.

Принимая это во внимание, достаточно сопоставить кодируемый сигнал с исходными данными и подсчитать расхождения в каждом из каналов. Выяснится, что в шестом канале будет наименьшее число ошибок (одна).

Ответ: 6

Задача 2.4.3. Тьюринг-младший (30 баллов)

Темы: кодирование-декодирование, анализ кода.

Условие

При широком развитии интернета и мобильной связи любительская радиосвязь до сих пор процветает и остается распространенным хобби. Люди с разных концов планеты с помощью собственных радиостанций связываются друг с другом, чтобы пообщаться голосом или обменяться короткими приветствиями на азбуке Морзе.

Невольно с этим хобби соприкоснулся и Алик, найдя на чердаке ламповую радиостанцию и собранную дедушкой телеграфную клавиатуру. Время пощадило технику, но не клей, на котором держались наклейки клавиш — они отпали от клавиатуры и лежали отдельно. В результате поиска и ревизии наклеек выяснилось, что клавиатура может отправлять заглавные латинские символы, цифры, пробел и некоторые знаки препинания (точка, запятая, восклицательный и вопросительный знак).

Алик решил восстановить раскладку клавиатуры. Как смог, он разложил наклейки, после чего стал набирать сообщения и смотреть, что именно отправляет клавиатура. Напишите программу, которая поможет Алику поскорее разобраться с раскладкой.

Задача интерактивная, поэтому программа взаимодействует с проверочной системой посредством стандартных потоков ввода и вывода. Иными словами, после отправки сообщения программа должна очистить буфер (выполнить `flush`) и считать ответ от системы (прочитать строку).

Итак, программа отправляет сообщение в раскладке Алика и получает в ответ от системы фактически отправленное. Например, программа отправит `HELLOIAMALIK`, после должна прочитать строку из потока ввода, где получит, например, `URYVBVNZNIVX`. Затем можно вновь отправлять очередное сообщение. Регистр латинских символов важен.

Как только программа восстановит кодировку, она должна отправить сообщение с символом `@` в начале и через пробел привести такое сообщение, чтобы на выходе получилось `GADROM.TKSVYMUCHFRURCALL.URRST599.QTHIRKUTSK.OPISALIK.HW?`.

В случае правильного итогового ответа программа ответит `@ OK`, и следует штатно завершить программу. В ином случае проверочная система выставляет `WA`. Также при наличии в любом сообщении символа не из алфавита задачи (например, строчных букв), проверка посылки завершается досрочно с вердиктом `PE`.

Например, программа присылает: `@ TN QE BZ. GXF IL ZHPU SE HE PNYY. HE EFG 599. DGU VEXHGFX. BC VF NYVX. UJ?`. Если это правильный ответ, проверочная система ответит `@ OK`, иначе завершит программу принудительно.

В пределах каждого теста нужно справиться с восстановлением кодировки не более, чем за 20 запросов. Если больше — решение отклоняется с `WA`. Если участник справился за минимально возможное число запросов, он получит за соответствующий тест максимальный балл, в противном случае — лишь половину.

Эта задача проверяет навык анализа кодировки, что является частью задачи заключительного этапа.

Примечания

- Каждое входящее и исходящее сообщение должно заканчиваться переносом строки.
- Очистка буфера делается следующим образом:
 - ◊ `fflush(stdout)` в C;
 - ◊ `fflush(stdout)` или `cout.flush()` в C++;
 - ◊ `sys.stdout.flush()` в Python;
 - ◊ `System.out.flush()` в Java.

Формат входных данных

Строки, состоящие из заглавных латинских символов, цифр, пробелов, точек, запятых, восклицательных и вопросительных знаков. Если отправляется итоговый ответ, строка должна начинаться со знака @ и пробела.

Формат выходных данных

Строки аналогичного формата.

Примеры

Стандартный ввод
HELLO I AM ALIK @ TN QE BZ. GXF IL ZHPU SE HE PNYU. HE EFG 599. DGU VEXHGFX. BC VF NYVX. UJ?
Стандартный вывод
URYUB V NZ NYVX @ OK

Тесты

<https://disk.yandex.ru/d/k52LF9wIEe7pbq>.

Решение

В задаче представлен тривиальный шифр замены с нестандартным алфавитом. Необходимо восстановить алфавит замены. Это можно сделать одним сообщением, содержащим полный алфавит: «ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.,!?»». Получив ответ от проверочной системы, нужно сопоставить символ сообщения с символом отправленного алфавита и сформировать итоговую посылку, ориентируясь на эти сопоставления.

Ограничение на 20 посылок в условии для частичного решения — намеренно оставленная ловушка.

Авторское решение задачи на языке Python 3.12 представлено по ссылке: <http://disk.yandex.ru/d/mSJEPZJZqp6MKw>.

Задача 2.4.4. Цифровая насадка (30 баллов)

Тема: автономное управление.

Условие

Следить за чем-то бывает не так просто, а когда следить нужно сразу за многими вещами — здесь справится не каждый. Но можно доверить эту задачу роботам.

На Белоптиччинской птицефабрике проводится работа по технологизации слежки за подрастающими цыплятами. Руководство фабрики обратилось за помощью к местному кружку робототехников, и они собрали «робо-маму». Она круглосуточно отслеживает и различает цыплят с помощью ИИ и машинного зрения. Не хватало лишь одной немаловажной части — «подружить» робота с системой управления птицефабрикой, написанной на языке COBOL задолго до основания кружка.

Напишите программу-драйвер для «робо-мамы», реализующую логику приема и ответа согласно заданного протокола. Система управления отправляет в драйвер одну из следующих команд:

- **REMEMBER name** — добавить в память цыпленка с заданным именем. Если такой цыпленок уже добавлен, ответить **ERROR EXIST**, иначе **OK**.
- **FORGET name** — удалить цыпленка с заданным именем из памяти. Если такой цыпленок уже удален, ответить **ERROR MISSING**, иначе **OK**.
- **WATCH name** — начать следить за цыпленком с заданным именем. Если такой цыпленок не существует, ответить **ERROR MISSING**, иначе **OK**, даже если слежение за цыпленком уже ведется.
- **UNWATCH name** — перестать следить за цыпленком с заданным именем. Если такой цыпленок не существует, ответить **ERROR MISSING**, иначе **OK**, даже если слежение за цыпленком не ведется.
- **LIST type** — перечислить цыплят в алфавитном порядке через запятую с пробелом. Например: 123, СНЕЕРУЗ, ТАПІ. Параметр **type** задает необходимый список: **ALL** — вывести всех цыплят в памяти, **WATCH** — всех, за которыми идет слежка.

Имя цыпленка (**name**) всегда представляет собой последовательность заглавных латинских букв и цифр. При попытке назвать цыпленка иначе нужно отвечать **ERROR NAME** в любой команде. Если переданная команда не соответствует протоколу, следует отвечать **ERROR SYNTAX**.

Эта задача проверяет навык работы с протоколами и их реализацией, что является частью задачи заключительного этапа.

Формат входных данных

Последовательность строк (до 200), в каждой из которых приведена отдельная команда.

Формат выходных данных

Такое же число строк, в каждой из которых приводится ответ на соответствующую команду.

Примеры

Стандартный ввод
WATCH DOKOP98Y
WATCH z
WATCH gejb
WATCH W9S
REMEMBER 31K882IR
WATCH 31K882IR
WATCH 4S9C9Q8
WATCH 31K882IR
WATCH 31K882IR
WATCH 31K882IR
...
Стандартный вывод
ERROR MISSING
ERROR NAME
ERROR NAME
ERROR MISSING
OK
OK
ERROR MISSING
OK
OK
OK
...

Тесты

<https://disk.yandex.ru/d/tdZA1mLOdrEuLQ>.

Решение

Задача предполагает прямолинейную реализацию заданного протокола и интерпретацию всех переданных во входных данных команд, в том числе ошибочных (очень важно разделять ошибки формата имени и формата сообщения). Для хранения информации о цыплятах предлагается использовать структуру типа «словарь».

Авторское решение задачи на языке Python 3.12 представлено по ссылке: <http://disk.yandex.ru/d/BfINPEVtKYTKfQ>.

Задача 2.4.5. Бегущая строка (20 баллов)

Тема: кодирование-декодирование.

Условие

Передавать изображения на расстояние научились сравнительно недавно, в конце XIX века, а первые телевизоры просуществовали недолго, быстро уступив место более привычным электронно-лучевым трубкам. Но идея, лежащая в основе первых телевизоров, проста и интересна, и такой телевизор можно собрать своими руками (например, как в этой статье: <https://nplus1.ru/material/2021/12/01/nipkow-avito>).

Представим механический телевизор Б-2 на основе диска Нипкова, рассчитанного на отображение кадра в 30 строк с частотой 12,5 кадров в секунду. В телевизор передается двоичный сигнал частотой 22,5 кГц, задающий яркость светодиода (соответственно, в двух значениях — максимальная и минимальная).

Предлагается файл (<https://disk.yandex.ru/d/VJHdZns97iwaLg>) с сигналом, передаваемым в такой телевизор. Обусловимся, что прием сигнала начинается одновременно с движением самой верхней прорези от левого угла кадра, вращение идеально синхронизировано с передаваемым сигналом, а трапецевидным искажением изображения можно пренебречь.

Восстановите сообщение, содержащееся в видеоизображении, и приведите его в ответе. Оно состоит из букв русского алфавита и знаков препинания. Регистр важен.

Чтобы файл не был таким большим, дублирующие кадры удалены, а сигнал сжат с помощью алгоритма RLE. Для наглядности в качестве максимальной и минимальной яркости используются символы «*» и «.» соответственно. Так, 3*1.3*2.. будет декодировано в ***.***..

Эта задача проверяет навык работы с кодировками и особенностями канала связи, что является частью задачи заключительного этапа.

Решение

RLE — способ сжатия через кодирование непрерывно повторяющихся последовательностей. Следовательно, для декодирования необходимо продублировать символы «*» и «.» указанное число раз. Для простоты решено не использовать отрицательные индексы, кодирующие неповторяющиеся последовательности.

Далее замечаем, что в задаче идет речь о строчной развертке сигнала. Так как частота передачи составляет 12,5 кадров в секунду при 30 строках, всего за секунду передается 375 строк. Частота входного двоичного сигнала указывает на число измерений в секунду, которое составляет 22500 Гц. Поделив это число на 375, получаем 60 — это горизонтальное разрешение, то есть число точек изображения (и соответственно символов) в одной строке.

Следовательно, разрешение кадра составляет 60 на 30 точек. Зная это, можно разбить декодированную по RLE последовательность символов на строки по 60 символов и отделить последовательности по 30 строк, формируя цепочку кадров. Получение искомого текста из нее осуществляется визуально.

Ответ: сообщение: «Радио — это улей, в котором слышится жужжание всего мира!».

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

На заключительном этапе в командном туре каждая команда профиля Технологии беспроводной связи работает с экспериментальными стендами, входящими в одноименный образовательный комплекс. Его программно-аппаратная платформа позволяет взаимодействовать с адаптивными алгоритмами и точностью передачи сигнала.

В современном мире все большее значение имеют технологии передачи больших зашифрованных массивов данных, в рамках которых при обмене сравнительно небольшими динамически изменяющимися ключами можно осуществлять доверенное шифрование и дешифрование. Задачи профиля связаны с актуальными задачами систем связи, включая:

- вопросы помехоустойчивого кодирования;
- передачи информации в условиях шумов;
- работы с различными форматами данных;
- разработки адаптивной системы слежения;
- создание собственных протоколов связи.

Ключевые области применения связаны с космосом, промышленным интернетом вещей, подводной и мобильной робототехникой, каналами связи для роевых устройств.

Задачи, стоящие перед участниками, ежегодно меняются, вынуждая их осваивать, а не запоминать технологию — всегда в реальных меняющихся условиях принимать решения и создавать собственные протоколы, которые могут работать при различных шумах, препятствиях и разных траекториях движения автономных устройств друг относительно друга.

Задачи профиля на всех этапах составлены таким образом, что для их решения требуются знания не только школьного уровня, но и углубленные знания в области программирования, математики и геометрии, а также азы по помехоустойчивому кодированию. От этапа к этапу увеличивается как сложность задач, так и их специфика.

Второй этап (дистанционный командный) — отборочный этап, задачи которого подобраны таким образом, чтобы объективно проверить индивидуальные знания участников и косвенно оценить личный вклад участников в результат командной работы. Это не только отборочное соревнование, но прежде всего — введение в профиль. То есть его содержание (задачи, динамика) несет образовательную составляющую, причем не только в рамках подготовки к заключительному этапу, но и для развития общих навыков (командное взаимодействие, работа со сложными моделями, информационный поиск и нестандартное мышление).

Второй этап профиля Технологии беспроводной связи включает в себя шесть командных задач. Все они сочетают в себе математику и информатику. Для решения требуются не только школьные знания, программные умения и навыки их применения в решении новых задач, но и факультативные знания в области программирования, математики и геометрии. Методические рекомендации к данному этапу позволяют очертить область знаний и навыков для самостоятельного изуче-

ния.

Кроме того, задачи второго этапа направлены на развитие элементов научного исследовательского подхода и навыков командной работы. Здесь участники формируют команды и загружают общее командное решение. Функций у этого этапа несколько:

- отбор команд из трех – пяти участников, способных решать сложные задачи;
- ознакомление участников с математическими и физическими концепциями, на которых будет построена задача заключительного этапа;
- предъявление командам требований к уровню программирования, идентичных тем, что будут предъявлены им на заключительном этапе.

Во время второго отборочного этапа участники решают задачи, которые отражают часть большой командной задачи заключительного этапа, и знакомятся с базовыми концепциями — работают с кодами, потоками и форматами данных, протоколами. Ниже представлена схема связей тем задач второго тура и навыков, необходимых для решения задач заключительного этапа.



Рис. 3.2.1. ТБС Схема связей тем задач

Анализ данных

Информация — главный ресурс, с которым участники работают на этом профиле, и важно уметь анализировать и обрабатывать ее в зависимости от задачи. При работе с этим разделом важно обращать внимание на формат данных, так как их представление может быть крайне разнообразно.

Кодирование-декодирование

Во время решения задачи заключительного этапа потребуется реализовать алгоритмы кодирования и декодирования, и особенности этих процессов проявлены в задачах этого раздела, начиная с использования уже готовых кодов, заканчивая разработкой собственных алгоритмов для частных проблем передачи данных.

Алгоритмы

Задача заключительного этапа предполагает написание программного кода, и здесь важную роль играет навык разработки алгоритмов, равно как и поиска подходящих типовых. На проработку этих навыков и рассчитаны задачи раздела «Алгоритмы». При работе с ними важно делать акцент на информационном поиске и умении выявить типовую подзадачу.

Анализ кода

Правильный выбор кода, в зависимости от задачи, дает половину решения. Эти задания позволят рассмотреть коды в разных аспектах, увидеть типичные проблемы и, как следствие, более осознанно и осмысленно подойти к этой части задачи заключительного этапа.

Канал связи

Одна из составляющих процесса передачи данных — канал связи, обладающий своими характеристиками. Этот раздел содержит задачи, связанные с анализом этих характеристик, в основном отраженных в вероятностной форме. Поэтому при решении понадобятся, как минимум, базовые знания теории вероятностей.

Автономное управление

Часть задачи заключительного этапа составляет разработка алгоритма автономного управления спутником, качество которого напрямую влияет на общий результат, поэтому важно понимать принципы автономности и обладать навыками создания самостоятельных алгоритмов. Хотя в некоторой степени это характерно для всех заданий, задачи этого раздела направлены именно на автономику.

Все задачи второго отборочного этапа решаются на платформе Яндекс.Контест. В заданное время всем участникам открывается доступ к очередному набору задач (их условиям и проверочной системе). Участники должны написать программу на одном из языков программирования: Python 3.12, Java 17, C++20, C17, и загрузить ее текст на сервер. На сервере выполняется тестирование загруженной программы: ее ответы сверяются с ответами эталонного решения, составленного авторами. Их решение требует наличие базовых навыков программирования, поскольку это является необходимой частью задания. Ввод данных в программу осуществляется через стандартный ввод (`stdin`), вывод данных — через стандартный вывод (`stdout`). Случайные числа, используемые для генерации тестов, являются псевдослучайными. Все проверки программ всех участников производятся на одинаковых данных.

Командная работа начинается именно во время второго этапа. Помимо общих механизмов, для профиля создано условие на командное взаимодействие — ограниченное число попыток на каждую задачу для команды. Это приводит к необходимости уже на начальном этапе договариваться о стратегии сдачи решений общекомандно и обозначать свою позицию по отношению к каждой задаче внутри команды — берет ли член команды ее на себя, пишет ли варианты решений, или передает ответственность другим.

Так, задачи разделяются между участниками согласно их сильным и слабым сторонам, что создает предпосылки к формированию ролей. Учет общекомандных попыток ведется посредством внешнего бота, предоставляющего текущую информацию о командных баллах и оставшихся командных попытках. Бот оживляет и делает осмысленным соревнование во время второго тура — оно становится проявленным и работает на повышение мотивации.

На протяжении всего второго тура на вебинарах поднимаются темы совместных ресурсов (Яндекс.Диск, Github) для структурирования командной работы. На вебинарах не всегда возможно присутствие всех членов команды — и это порождает отдельный процесс по синхронизации информации и понимания. В конце второго тура командам предлагается совместная рефлексия с выявлением слабых и сильных сторон у каждого члена команды, определением белых пятен в знаниях и навыках у каждого и распределением нерешенных задач для дорешивания.

3.2.1. Командные задачи

На заключительном этапе профиля Технологии беспроводной связи будут предложены задачи построения сетей и обеспечения связи в условиях, когда одновременно с ростом числа подключаемых устройств и объемов трафика повышаются требования к надежности и производительности систем.

Прежде всего, второй этап — командный, поэтому оцениваться будет общий результат. Главная цель здесь — формирование команды. Чем лучше будет выстроена командная работа на втором этапе, тем больше шансов на победу будет в каждом из этапов. Обратите внимание на особенность профиля — уже со второго этапа высока роль капитана, от его эффективных действий и распределения функций между членами команды будет зависеть успех выступления команды на втором туре. Общайтесь, учитесь эффективно распределять работу, ищите сильные стороны каждого участника, наращивайте общекомандные навыки — и коммуникативные, и профессиональные. Решайте задачи максимально эффективно и результативно.

Задачи данного этапа знакомят с базовыми концепциями. В них будут затронуты следующие темы:

- анализ данных,
- кодирование – декодирование,
- алгоритмы,
- анализ кода,
- канал связи,
- автономное управление.

Все задачи требуют написания программного кода на одном из следующих языков программирования:

- Python 3.12,
- Java 17,
- C++20,
- C17.

Можно работать с любой удобной IDE (Visual Studio Code, Spyder и т.д.) или

онлайн-средой (Repl.it, Google Colab и т. д.). Учитывайте, что на платформе Яндекс.Контест во всех языках доступны только стандартные модули без дополнительных библиотек.

Командные задачи второго этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контест: <https://contest.yandex.ru/contest/69926/enter/>.

Задача 3.2.1.1. Загадки бумажных болот (10 баллов)

Темы: анализ кода, алгоритмы, навык работы с кодировками.

Условие

Главному герою фантастической игры Семену не повезло — он застрял в бюрократической параллельной вселенной, в одной далекой стране на альтернативной Земле. В этой стране для регулирования паролей в аккаунтах государственных сервисов используется особая директива — Перечень регламентов утверждения действительности информационной кодировки (ПРУДИК). Каждый государственный сервис каждому новому аккаунту предписывает набор правил ПРУДИКа, по которым пользователю нужно сочинить новый пароль и отправить его бумажной почтой. Чтобы попасть в родную вселенную, Семену предстоит пройти чудовищное испытание — зарегистрироваться на всех онлайн-сервисах и подобрать к ним пароли, соответствующие поставленным правилам Перечня.

ПРУДИК состоит из пронумерованного набора правил. Каждое правило определяет ограничение на необходимый пароль.

1. Пароль должен содержать не менее 5 символов.
2. Пароль должен содержать арабские цифры.
3. Пароль должен содержать заглавную букву английского алфавита.
4. Пароль должен содержать один из специальных символов: ~!@#\$%^&*.
5. Сумма всех арабских цифр в пароле должно равняться 25.
6. Пароль должен содержать название месяца на английском языке с заглавной буквы.
7. Сумма всех арабских и римских¹ чисел в пароле должна равняться 35.
8. Пароль должен содержать римскую цифру.
9. Перемножение всех римских чисел¹ в пароле должно равняться 35.
10. Каждое слово² в пароле должно заканчиваться на .ru.
11. Каждое слово должно содержать не более 7 символов.
12. Пароль не должен содержать пробелов.
13. Каждое слово должно начинаться с большой буквы.
14. Пароль должен заканчиваться восклицательным знаком!
15. Пароль должен начинаться с символа \$.
16. Пароль является палиндромом (без учета пробелов и регистра).
17. Пароль содержит корректно расставленные скобки: () [] { } < >.

¹ Римские числа — это непрерывная последовательность римских цифр, приведенных в порядке невозрастания их мощности.

² Слово — это последовательность символов, разделенных одним и более пробелом.

Реализуйте бота, который выполняет проверку соответствия сообщений ПРУ-ДИКу и подсказывает, подходит ли придуманный пароль к очередному сервису.

Формат входных данных

- *1, 2, 3, 4 — заявляет набор правил, которые необходимо проверять.
- ? **проверь меня** — проверить сообщение на соответствие правилам. Сообщение рассматривается без вопросительного знака и пробела в начале.
- # — закончить работу.

Формат выходных данных

- NO 1, 2, 3, 4 — ответ на проверку, если сообщение не соответствует правилам. Через запятую приводятся нарушенные правила в порядке возрастания номера.
- OK — ответ на все остальные сообщения, в том числе на проверку в случае полного соответствия.

В начале работы система отправляет сообщение и ждет от ответа. Обработав завершающее сообщение, решение должно штатно завершиться.

Тесты

https://disk.yandex.ru/d/BxyK_E8L3mqlEg

Решение

В бесконечном цикле проверяем входящие команды. Проверка строки происходит посредством последовательной проверки каждого правила. Большинство правил проверяются с помощью регулярных выражений.

Последовательности римских чисел выделяются регулярным выражением, после чего делятся на отдельные числа, основываясь на том, что римское число представляет собой последовательность цифр от большего к меньшему, так как в условии исключено правило вычитания. Например, последовательность XIIVIII будет разделена на числа XII и VIII.

Правильность скобочного выражения определяется с помощью стека. По мере продвижения по строке, добавляем в стек считанные открывающие скобки. При чтении закрывающей скобки проверяется, соответствует ли ей скобка в голове стека. При совпадении она вынимается из стека, иначе проверка завершается с отрицательным вердиктом. Кроме того, отрицательный вердикт будет при закрывающей скобке при пустом стеке, а также непустом стеке по завершении чтения строки.

Авторское решение на языке Python 3.12 приведено в файле TWC-prudik.py: <https://disk.yandex.ru/d/6Ha2By4yMqaKXw>.

Критерии оценивания

Это интерактивная задача, значит, программа взаимодействует с проверочной системой посредством стандартных потоков ввода и вывода. Иными словами, после отправки сообщения программа должна очистить буфер (выполнить `flush`) и считать ответ от системы (прочитать строку). В случае некорректного ответа проверка прерывается с вердиктом PE (Presentation Error).

Задача 3.2.1.2. Кошмарный писк радиофизика (26 баллов)

Темы: канал связи, навык работы с моделями передачи сигнала.

Условие

Летучие мыши Порфирий и Настасья мирно жили в уютной пещере и общались между собой ультразвуком. Но в какой-то момент они перестали нормально слышать друг друга — эхо стало оглушительно громким, и различить, где сигнал, а где эхо, уже не получается. Оказаться в кошмарном сне радиофизика неприятно, но нужно выходить из этой ситуации. А так как радиофизик спит беспокойно, но крепко, помогите мышам адаптироваться к новым физическим законам выдуманного мира, а именно: сопоставить передаваемый и принимаемый звук.

Пусть пещера имеет форму выпуклого многоугольника, летучих мышей представим в виде точечных излучателя и приемника. Излучатель передает сигнал в течение 3 с. Прием начинается одновременно с передачей и также длится 3 с. Сигнал проходит по воздуху со скоростью 343 м/с без затухания. Отражение также происходит без затухания, но только один раз. Частота дискретизации составляет 24 кГц. На приемнике принятые сигналы суммируются алгебраическим сложением уровней. Наложением звука на отражение пренебречь. Гарантируется, что излучатель и приемник находятся строго внутри многоугольника. Считать, что сигнал может проходить через излучатель и приемник.

Необходимо написать программу для приемника, которая восстанавливает принятый сигнал.

Формат входных данных

В первой строке — четыре вещественных числа через пробел, пары координат (абсцисса и ордината) излучателя и приемника соответственно. В следующей строке указывается натуральное число N , число точек в многоугольнике. Далее идет N строк, в каждой из которых указывается пара вещественных чисел — абсцисса и ордината очередной точки многоугольника. Далее идет 72 000 строк, в каждой — вещественное число, значение передаваемого сигнала в условных единицах. Точки многоугольника приводятся по часовой стрелке, все координаты — в метрах.

Формат выходных данных

72 000 строк, в каждой — вещественное число, значение принятого сигнала в условных единицах.

Примеры*Пример №1*

Стандартный ввод
2.5 2.5 -2.5 2.5 4 -5 5 5 5 5 -5 -5 -5 2.811570446214362 35.26305252415367 31.513343289467844 ... 0.0
Стандартный вывод
0.0 0.0 0.0 ... 21.263634409701194 10.45558991528608 13.266212134182132 ... 0.0

Тесты

<https://disk.yandex.ru/d/CtNZryVHnW1MUQ>.

Решение

Задача решается по следующему алгоритму.

- Вычисляем прямое расстояние от передатчика до приемника.
- Выясняем для каждой стороны многоугольника, может ли отраженный от передатчика сигнал попасть на приемник, и если да, то какое расстояние придется преодолеть сигналу:
 - ◊ Строим перпендикуляры от передатчика и приемника до прямой, на которой лежит рассматриваемая сторона многоугольника.
 - ◊ Проведя отрезок от передатчика до приемника, получим трапецию, составленную из этого отрезка (назовем его верхним основанием), перпендикуляров и отрезка между перпендикулярами (назовем его нижним основанием).
 - ◊ На нижнем основании должна лежать точка отражения сигнала передатчика в приемник, согласно закону физики: угол отражения равен углу падения.

- ◇ Проведя отрезки от передатчика и приемника до точки отражения, получим три треугольника внутри трапеции.
- ◇ Известно, что два из трех треугольников являются прямоугольными, а также что углы, образованные точкой отражения, равны (по законам физики). Из этого следует, что они подобны. Благодаря свойствам подобия можно вычислить стороны, разделенные точкой отражения. С помощью значений этих сторон и длин перпендикуляров можно вычислить гипотенузы по теореме Пифагора. Сумма этих гипотенуз является расстоянием, которое сигналу нужно пройти от передатчика до приемника, если он отразится от рассматриваемой стороны многоугольника.
- ◇ Так как нижнее основание может выходить за пределы стороны многоугольника, необходимо проверить, что точка отражения действительно лежит на этой стороне. Ее координаты можно вычислить, решив систему уравнений, составленную из уравнения прямой и формулы Евклидова расстояния.
- После нахождения длин всех маршрутов от передатчика до приемника определяем значение сигнала, который примет приемник:
 - ◇ Вычисляем время прихода сигнала до приемника по каждому маршруту.
 - ◇ Для каждого значения сигнала передатчика вычисляем такт, в который он дойдет по всем маршрутам.
 - ◇ Складываем сигналы, пришедшие в один и тот же такт. Итоговая сумма сигналов дает ответ.

Авторское решение задачи на языке Python 3.12 приведено в файле `TWC-soundroom.py`: https://disk.yandex.ru/d/tM6_nIUSCIHNTw.

Критерии оценивания

Решение принимается, если среднее квадратичное отклонение между решением и авторским ответом составит менее 10 условных единиц.

Задача 3.2.1.3. Работа из-под импульса, подзадача 1 (11 баллов)

Темы: кодирование – декодирование, навык работы с протоколами, автономное управление.

Условие

Белоптичкинский радиоловительский клуб славится находчивыми специалистами. Недавно они обнаружили, что, скрестив катушку индуктивности, конденсатор и секретный компонент (назначение которого не понимают даже сами изобретатели), можно собрать очень эффективный цифровой приемник, работающий без батареек, чисто на входящем сигнале. Точнее, на восходящих фронтах сигнала (когда значение меняется из нижнего уровня в высокий).

Правда, высокая эффективность дается высокой ценой. Чтобы приемник в принципе начал работать, на него требуется подать 6 восходящих фронтов в течение 15 тактов. Прием сигнала начинается на следующий такт после включающего фронта. Мало того, для поддержки питания необходимо, чтобы за последние 15 тактов

было минимум 4 восходящих фронта. В ином случае при очередном такте без фронта приемник выключится, не приняв значение в этом такте. В самом начале на приемнике установлен низкий уровень сигнала.

Дано: тактированный сигнал, который будет передан на приемник. Определите, какое сообщение будет с него получено.

Формат входных данных

Одна строка, произвольное количество нулей и единиц, принимаемый сигнал.

Формат выходных данных

Одна строка, произвольное количество нулей и единиц, считанное сообщение.

Примеры

Стандартный ввод
1010101010111010010000011010101010111111101
Стандартный вывод
1101001000001111111101

Тесты

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

Решение

Для хранения последних 15 значений входной последовательности используем циклический буфер, который изначально заполняем пустыми значениями. Считываем входную последовательность, а затем посимвольно добавляем ее в буфер, считая восходящие фронты и включая/выключая передачу по необходимости.

Считается, что до входной последовательности были нули, поэтому в начале добавляем ноль в последовательность, чтобы учесть восходящий фронт в случае, если последовательность начинается с единицы.

Также учитывается, что ноль, который находится перед обрабатываемыми 15 значениями, может образовывать фронт, если первым значением в буфере является единица.

Авторское решение задачи на языке Python 3.12 приведено в файле `TWC-pulsedriver-a.py`: <https://disk.yandex.ru/d/87dAE18CcbkZqQ>.

Задача 3.2.1.4. Работа из-под импульса, подзадача 2 (15 баллов)

Темы: кодирование – декодирование, навык работы с протоколами, автономное управление.

Условие

Напомним, цифровой приемник включается, когда на него в течение 15 тактов приходит 6 восходящих фронтов. Прием сигнала начинается на следующий такт после включающего фронта и продолжается до тех пор, пока в течение 15 тактов приходит минимум 4 восходящих фронта. В ином случае при очередном такте без фронта приемник выключится, не приняв значение в этом такте. В самом начале на приемнике установлен низкий уровень сигнала.

Теперь решите обратную задачу: спроектируйте такой сигнал, чтобы приемник получил заданную двоичную последовательность.

Формат входных данных

Одна строка, произвольное количество нулей и единиц, передаваемое сообщение.

Формат выходных данных

Одна строка, произвольное количество нулей и единиц, сформированный для передачи сигнал.

Примеры

Пример №1

Стандартный ввод
1101001000001111111101
Стандартный вывод
1010101010111010010000011010101010111111101

Тесты

https://disk.yandex.ru/d/b_c9SUX-7OT_Yg.

Решение

Выполняется операция, обратная предыдущей — формируется такой сигнал, чтобы приемник из первой задачи сформировал строку, идентичную ответу. Соответственно, сигнал формируется из входной строки побитово с подсчетом фронтов, как в первой задаче. Если после очередного переданного бита число восходящих

фронтов становится меньше необходимого, передается комбинация 01 в количестве, достаточном для включения, после чего передача возобновляется. Проверка задачи осуществляется программно, поэтому способ включения приемника может быть избыточным и предполагает лишь соответствие условию задачи.

Авторское решение задачи на языке Python 3.12 приведено в файле `TWC-pulsedriver-b.py`: <https://disk.yandex.ru/d/eXWPBbrndnOnFg>.

Задача 3.2.1.5. Не Фурье (29 баллов)

Темы: анализ данных, кодирование – декодирование, навык анализа и формирования сигналов.

Условие

В Белоптичкинском кружке робототехников готовятся к районной конференции по киберфизике. На один из докладов школьники подготовили робота-попугая, очень похожего на природный прообраз. Умеет даже разговаривать, запоминая и повторяя звуки... Но беда в том, что за полнедели до выступления из строя вышел контроллер, отвечающий за функцию звукоподражания. Юные гении не растерялись и смогли собрать аналогичную схему. Но в ее основе лежит более слабый контроллер, поэтому его дополнили тремя генераторами звука. Осталось написать драйвер.

На его вход передается сигнал длиной до 500 квантов, который необходимо повторить посредством активации трех генераторов в определенном порядке. Для каждого из генераторов известен формируемый им сигнал и время восстановления между завершением генерации и повторной активацией. Все сигналы передаются в квантованном виде с одинаковой частотой дискретизации. Наложение сигналов производится сложением уровней.

Реализуйте описанный драйвер.

Формат входных данных

Четыре блока с передаваемыми данными.

Первый блок — в первой строке натуральное число N , длительность сигнала в квантах времени.

Далее идет N строк, в каждой из которых приводится вещественное число, уровень сигнала в соответствующий квант времени.

Следующие три блока устроены одинаково и описывают генерируемые формы сигналов. Первая строка — два натуральных числа N и M через пробел, длительность формы и время восстановления в квантах времени. Далее идет N строк, в каждой из которых приводится вещественное число, уровень генерируемой формы в соответствующий квант времени.

Формат выходных данных

В первой строке — натуральное число N , количество активаций генераторов.

Далее идет N строк, в каждой из которых идет пара целых чисел через пробел. Первое число — номер кванта времени (нумерация с 0!), второе — номер активируемого генератора (1, 2, 3).

Примеры

Пример №1

Стандартный ввод
48 28.384381050597984 41.03454979919398 64.56745071160744 86.06538001102359 ... 26.94524
Стандартный вывод
5 0 2 9 3 16 1 25 2 32 3

Тесты

<https://disk.yandex.ru/d/ja4TN1YJdA0mvQ>.

Решение

В основе авторского решения лежит рекурсивный перебор, который заранее исключает неподходящие варианты, за счет чего обеспечивается быстрота решения.

Рекурсивно посмотрим каждый квант заданного сигнала, выполняя следующие действия:

- Рассчитать оценку среднеквадратичной невязки между текущим составленным решением и эталонным сигналом.
- Если значение оценки превышает 10, то следует рекурсивно вернуться на квант назад и выбрать другой вариант.
- Если рассматривается последний квант, и оценка невязки не превышает 10, то это и есть решение, нужно закончить перебор и вывести ответ.
- Произвести перебор вариантов запуска генераторов.
- Исключить такие варианты, в которых генераторы еще не закончили работу или находятся на восстановлении.
- Рассчитать изменения среднеквадратичной невязки для каждого варианта.
- Отсортировать варианты по возрастанию оценки среднеквадратичной невязки и выбрать вариант с наименьшей невязкой.

- Перейти к следующему кванту с учетом выбранного варианта.

Авторское решение задачи на языке Python 3.12 приведено в файле `TWC-unfourier.py`: <https://disk.yandex.ru/d/Zqrtrv9Lf9j8IQ>.

Критерии оценивания

Решение принимается с оценкой в 50% за тест, если среднеквадратичная невязка относительно входного сигнала составит не более 50 единиц. Для полного балла необходима невязка не более 10 единиц.

Задача 3.2.1.6. Кибер-наседка (9 баллов)

Темы: автономное управление, навык работы с протоколами и их реализацией.

Условие

Следить за чем-то бывает не так просто, а когда следить нужно сразу за многими вещами — здесь справится не каждый. Но можно доверить эту задачу роботам.

На Белоптичкинской птицефабрике проводится работа по технологизации слежки за подрастающими цыплятами. Руководство фабрики обратилось за помощью к местному кружку робототехников, и они собрали «робо-маму». Она круглосуточно отслеживает и различает цыплят с помощью ИИ и машинного зрения.

В прошлом году ребятам удалось «подружить» робота с системой управления птицефабрикой, написанной на языке COBOL задолго до основания кружка. Теперь же умные техники успели съездить на заключительный этап НТО по профилю Технологии беспроводной связи и поняли, как можно улучшить программу и реализовать новый режим работы, необходимый птицефабрике.

Напишите программу-драйвер для обновленной «кибер-мамы», реализующую логику приема и ответа согласно заданному протоколу. Система управления отправляет в драйвер одну из следующих команд:

- **REMEMBER name priority** — добавить в память цыпленка с заданным именем и приоритетом слежения (целое число от 1). Если такой цыпленок уже добавлен, ответить **ERROR EXIST**, иначе — **OK**.
- **FORGET name** — удалить цыпленка с заданным именем из памяти. Если такого цыпленка нет в памяти, ответить **ERROR MISSING**, иначе — **OK**.
- **WATCH name** — начать следить за цыпленком с заданным именем. Если такой цыпленок не существует, ответить **ERROR MISSING**, иначе — **OK**, даже если слежение за цыпленком уже ведется.
- **UNWATCH name** — перестать следить за цыпленком с заданным именем. Если такой цыпленок не существует, ответить **ERROR MISSING**, иначе — **OK**, даже если слежение за цыпленком не ведется.
- **LIST type** — перечислить цыплят в алфавитном порядке через запятую с пробелом. Например: **123, СНЕЕРУЗ, ТАПІ**. Параметр **type** задает необходимый список: **ALL** — вывести всех цыплят в памяти, **WATCH** — всех, за которыми идет слежка.

- `ECOMODE priority` — включить режим экономии энергии с заданным приоритетом. Приоритет 0 выключает этот режим. Ответить `OK`.

Режим экономии работает следующим образом: все цыплята, приоритет для которых ниже заданного, больше не отслеживаются. При этом драйвер продолжает принимать команды на измерение слежения за цыплятами, но лишь запоминает необходимое состояние, при этом не начиная следить за теми, кто не подходит по приоритету.

Если переданная команда не соответствует протоколу, следует отвечать `ERROR SYNTAX`.

Имя цыпленка (`name`) всегда представляет собой последовательность заглавных латинских букв и цифр. При попытке назвать цыпленка иначе нужно отвечать `ERROR NAME` в любой команде.

Приоритет всегда является целым положительным числом (0 допустим только в команде `ECOMODE`). При попытке прописать некорректный приоритет нужно отвечать `ERROR PRIORITY`.

При выдаче ошибок система отдает приоритет синтаксическим, после параметрическим, и только затем семантическим.

Формат входных данных

Строки с командами, перечисленными в условии.

В одной строке передается только одна команда. Но из-за особенностей драйвера могут встречаться лишние пробелы в начале и конце строки или между словами, а также пустые строки. Их нужно игнорировать.

Формат выходных данных

Строки с ответами на команды в соответствии с условием.

Примеры

Стандартный ввод
LIST 874505 REMEMBER LZLH 1893555056 REMEMBER LZLH 400606675 ECOMODE 0 REMEMBER OUMB4 2016754536
Стандартный вывод
ERROR SYNTAX OK ERROR EXIST OK OK

Тесты

<https://disk.yandex.ru/d/LJtJbreuMlXR9g>.

Решение

Программа считывает входные данные построчно до тех пор, пока не будет встречен маркер конца файла (EOF).

Для хранения записей о цыплятах используется словарь, в котором ключи — имена цыплят, значения — структура со значениями: приоритет и флаг отслеживания цыпленка в данный момент. Логика решения преимущественно реализует операции над словарем. Рекомендуется сортировать ключи словаря в алфавитном порядке. Если структура данных это не предусматривает, можно выделить отдельный массив для хранения сортированных ключей.

Для функции отображения отслеживаемых цыплят выполняем проход по словарю, для каждого цыпленка смотрим приоритет (если включен его учет) и флаг отслеживания.

При считывании команд тщательно проверяется синтаксис команды, наличие аргументов (не больше, не меньше), их корректность. Также учитывается, что могут быть лишние пробелы в строке с командой, причем даже между аргументами.

Авторское решение задачи на языке Python 3.12 приведено в файле `TWC-cyber-hen.py`: <https://disk.yandex.ru/d/rciUWz8R09xS5A>.

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 формировании 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.Контеcт: <https://contest.yandex.ru/contest/72673/enter/>.

4.2.1. Информатика. 8–11 классы

Задача 4.2.1.1. Две площадки (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

Научный институт проектирует полигон для испытания оборудования беспроводной связи. Для испытаний на полигоне требуется разместить две прямоугольные площадки, имеющие размеры $a_1 \times b_1$ и $a_2 \times b_2$. Они могут соприкасаться, но не должны пересекаться. Полигон также должен иметь форму прямоугольника, и стороны площадок должны быть параллельны или перпендикулярны сторонам полигона.

Проектировщики хотят сделать полигон минимальным по площади, но так, чтобы на нем поместились обе площадки. Напишите программу, которая найдет наименьшую возможную площадь полигона по заданным размерам площадок.

Программа должна решить задачу для t независимых наборов входных данных.

Формат входных данных

В первой строке на вход подается единственное натуральное число t — количество наборов входных данных ($1 \leq t \leq 10\,000$). Далее записаны сами наборы входных данных. Каждый набор состоит из двух строк. В первой строке через пробел записаны два натуральных числа a_1 и b_1 — размеры первой площадки. Во второй строке аналогично записаны еще два числа a_2 и b_2 — размеры второй площадки. Все числа не превосходят 10 000.

Формат выходных данных

Для каждого набора входных данных выведите в единственной строке ответ — минимальную площадь полигона.

Примеры*Пример №1*

Стандартный ввод
2
6 4
3 5
4 2
3 2

Стандартный вывод
42
14

Примечание. Рисунки ниже поясняют тестовые случаи.

**Решение**

Обозначим стороны первого прямоугольника за a_1 и b_1 , а второго — за a_2 и b_2 . Прямоугольники можно соединить по сторонам четырьмя способами: a_1a_2 , a_1b_2 , b_1a_2 и b_1b_2 . В результате получится прямоугольник, где длина одной стороны будет максимум из длин соединяемых сторон, а длина второй стороны будет суммой длин оставшихся сторон. В программе надо перебрать четыре указанных способа и выбрать из них лучший.

Пример программы-решения

Ниже представлено решение на языке Python.

Python

```

1 t = int(input())
2 for _ in range(t):
3     a1,b1 = map(int,input().split())
4     a2,b2 = map(int,input().split())

```

```

5     print(min((a1+a2)*max(b1,b2),
6               (b1+b2)*max(a1,a2),
7               (a2+b1)*max(a1,b2),
8               (a1+b2)*max(a2,b1)))

```

Критерии оценивания

Тест с номером 1 совпадает с тестом из условия задачи. Баллы за него не начисляются.

Успешное прохождение каждого теста с номерами 2–11 оценивается в 1 балл.

Задача 4.2.1.2. Кооперативная игра (12 баллов)

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

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

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Алиса, Алена и Алина играют в следующую игру. Перед игроками в ряд расположено n кучек камней, пронумерованных от 1 до n , $n \geq 3$. За a_i обозначим количество камней в i -й кучке. Сначала Алиса выбирает себе некоторую кучку камней, которая не является самой левой или самой правой. Обозначим номер кучки, выбранной Алисой, за k , $2 \leq k \leq n - 1$. Далее Алена выбирает любую кучку камней слева от кучки, выбранной Алисой, а Алина аналогично выбирает одну кучку справа. Обозначим номера кучек Алены и Алины за m и r , $1 \leq m < k < r \leq n$.

Выигрышем станет сумма камней в трех выбранных кучах. Игра является кооперативной, то есть цель игроков — максимизация совместного выигрыша.

Напишите программу, которая найдет номера кучек, выбранных Алисой, Аленой и Алиной. Она должна ответить на t независимых запросов во входных данных.

Формат входных данных

В первой строке на вход подается единственное натуральное число t — количество наборов входных данных ($1 \leq t \leq 10\,000$). Далее записаны сами наборы входных данных. Каждый набор состоит из двух строк. В первой строке записано одно натуральное число n — количество кучек с камнями ($3 \leq n \leq 200\,000$).

Во второй строке через пробел записано n натуральных чисел $a_1, \dots, a_n$ — количество камней в каждой кучке, $1 \leq a_i \leq 10^9$. Гарантируется, что сумма n по всем наборам входных данных не превосходит 200 000.

Формат выходных данных

Для каждого набора входных данных выведите с отдельной строке три числа k , m и r — номера кучек камней, которые возьмет Алиса, Алена и Алина, $1 \leq m < k < r \leq n$. **Обратите внимание**, что числа должны выводиться в требуемом порядке, а именно: сначала k , потом m и r .

Если возможно несколько правильных решений, то вывести можно любое из них.

Примеры

Пример №1

Стандартный ввод
3
3
5 1 4
5
1 2 3 4 5
5
2 2 2 3 1
Стандартный вывод
2 1 3
4 3 5
3 2 4

Примечания

В первом тестовом случае всего три числа. Алиса может выбрать только второе число.

Во втором тестовом случае Алиса выберет четверку, а Алена и Алина тройку и пятерку соответственно. Сумма будет равна 12, и улучшить этот результат нельзя.

В третьем тестовом случае Алиса выберет двойку, а Алена и Алина двойку и тройку соответственно. Сумма будет равна 7, и улучшить этот результат нельзя.

Решение

Очевидно, что сумма будет максимальной, если девочки выберут три самых больших числа. Поскольку Алиса выбирает число первой, она должна выбрать то число из этой тройки, которое находится в середине.

Таким образом, программа должна найти номера трех самых больших чисел, упорядочить эти номера по возрастанию и вывести сначала второй номер, потом первый и, наконец, третий.

В прилагаемой программе в пятой строке номера чисел сортируются в порядке возрастания их значений.

Пример программы-решения

Ниже представлено решение на языке Python.

Python

```

1  t = int(input())
2  for _ in range(t):
3      n = input()
4      x = [int(a) for a in input().split()]
5      x = sorted(list(range(len(x))), key = lambda i:x[i])
6      x = sorted(x[-3:])
7      print(x[1]+1,x[0]+1,x[2]+1)

```

Критерии оценивания

Тест с номером 1 совпадает с тестом из условия задачи. Баллы за него не начисляются.

Успешное прохождение каждого теста с номерами 2–13 оценивается в 1 балл.

В тестах с номерами 2–5 все числа a_i являются различными.

В тестах с номерами 2–9 сумма n по всем тестовым случаям не превосходит 20 000.

Задача 4.2.1.3. Удаленное управление обгонами (18 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

Антон пишет программу для управления гоночными машинками. Недавно он добавил модуль для выполнения обгонов и хочет его протестировать. Интерфейс модуля очень прост и содержит всего одну команду «обогнать». Выполняя эту команду, некоторая машинка обгоняет ровно одну другую машинку, двигающуюся впереди. Если перед машинкой, которая должна выполнить эту команду, нет другой машинки, то команда игнорируется.

Машинки двигаются по достаточно большому гоночному кругу, строго друг за другом с одинаковой скоростью, за исключением моментов обгонов. Это означает, что машинка, двигающаяся первой, не видит и не может обогнать машинку, которая едет последней. Все они пронумерованы последовательными числами от 1 до n . Каждая окрашена в один из трех цветов Российского флага, и количество машинок каждого цвета совпадает. Чтобы отправить команду на обгон, программа должна просто указать номер машинки, которая будет его выполнять.

Для эффектной демонстрации своего модуля Антон просит вас написать программу, которая составит такую последовательность обгонов, чтобы машинки на финише были упорядочены по цветам, как на флаге России сверху вниз. **Количество обгонов при этом должно быть минимальным.** В момент старта машинки выстроены по своим порядковым номерам.

Программа должна ответить на t независимых запросов во входных данных.

Формат входных данных

В первой строке на вход подается единственное натуральное число t — количество наборов входных данных ($1 \leq t \leq 333$). Далее записаны сами наборы входных данных. Каждый набор входных данных представлен в виде строки, состоящей из символов **w**, **b**, **r**. Гарантируется, что строка не пустая, и ее длина делится на 3. Также гарантируется, что сумма длин всех строк во входных данных не превосходит 1 000.

Будем считать, что нумерация символов в строке начинается с единицы. Тогда символ **w** в i -й позиции означает, что машинка с номером i белая. Аналогично символы **b** и **r** означают машинки синего и красного цветов.

Формат выходных данных

Ответ на каждый запрос во входных данных должен быть записан в **одной** строке в виде последовательности целых чисел. Первым числом в последовательности должно быть минимальное количество обгонов r . Далее должно быть записано еще r чисел — номера машинок, на которые подается команда об обгоне. Номера должны быть записаны в порядке, в котором они будут отправляться.

Если ответов с минимальным количеством обгонов несколько, то можно вывести любой.

Примеры

Пример №1

Стандартный ввод
3
rwbbbrw
wbr
wbrwbr
Стандартный вывод
7 2 6 6 6 6 3 4
0
3 4 4 5

Примечания

Последовательность обгонов в первом тестовом случае отображена на рис. 4.2.1. Эта последовательность не является единственной, возможны и другие ответы.

Во втором тестовом случае машинки уже стоят в нужном порядке, и обгоны не требуются.



Рис. 4.2.1

Решение

Фактически в этой задаче нужно реализовать произвольный алгоритм, выполняющий сортировку данных обменом соседних элементов. При этом требуется формировать специфическую последовательность номеров, поэтому алгоритмы из библиотеки не подойдут.

В прилагаемой программе формируется список из пар, состоящих из номера машинки и ее цвета. Далее выполняется алгоритм, похожий на сортировку вставками. Сначала выбираются все машинки белого цвета и перемещаются в начало последовательности. Далее такие же действия выполняются с синими машинками. На каждом шаге известна позиция, в которую надо переместить машинку, что упрощает реализацию алгоритма.

Пример программы-решения

Ниже представлено решение на языке Python.

Python

```

1  t = int(input())
2  for _ in range(t):
3      s = list(enumerate(input()))
4      ans = []
5      k = 0
6      for c in 'wb':
7          for i in range(len(s)):
8              if s[i][1]==c:
```

```

9         ans+=s[i][0]+1)*(i-k)
10        b = s[i]
11        for j in range(i,k,-1):
12            s[j]=s[j-1]
13        s[k]=b
14        k+=1
15        print(s)
16    print(len(ans),*ans)

```

Критерии оценивания

Тест с номером 1 совпадает с тестом из условия задачи. Баллы за него не начисляются.

Успешное прохождение каждого теста с номерами 2–19 оценивается в 1 балл.

Задача 4.2.1.4. XOR-машина и искусственный интеллект (30 баллов)

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

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

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 128 Мбайт.

Условие

На занятиях по схемотехнике Алиса разработала компьютер, способный выполнять два вида команд. Сегмент данных ее компьютера состоит из 128 двадцатиразрядных беззнаковых целых чисел, пронумерованных от 0 до 127. Программа состоит из команд, пронумерованных от 1 до n . Команда с номером i всегда записывает i -е число в сегменте данных, поэтому программа состоит не более, чем из 127 команд. Нулевое число в сегменте данных считается входной информацией и помещается в сегмент данных до начала выполнения программы.

Первая команда имеет вид $(i \text{ r } k \text{ m})$, где i — порядковый номер команды; **r** — ее обозначение (rotate); k — некоторое целое неотрицательное число, $k < i$; m — целое неотрицательное число, $0 \leq m \leq 19$. По этой команде из сегмента данных извлекается число под номером k , его биты циклически сдвигаются на m разрядов влево, результат записывается в сегмент данных под номером i .

Операция циклического сдвига влево может быть описана следующим образом. Пусть задано n -разрядное число $a_{n-1} \dots a_1 a_0$. Необходимо выполнить его циклический сдвиг влево на m бит. Выделим из числа m старших бит $a_{n-1} \dots a_{n-m}$ и запишем их в начало. Получим циклический сдвиг $a_{n-m-1} \dots a_1 a_0 | a_{n-1} \dots a_{n-m}$. Например, циклическим сдвигом числа 00101101 на три бита влево будет число 01101|001. Вертикальная черта поставлена для наглядности.

Вторая команда имеет вид $(i \text{ x } k \text{ t})$, где i — порядковый номер команды; **x** — ее обозначение (eXclusive or); k и t некоторые целые неотрицательные числа, $k < i$, $t < i$. По этой команде из сегмента данных извлекаются числа под номером

k и t , для них выполняется операция поразрядного исключающего «или», результат записывается в сегмент данных под номером i .

Булевская операция Хор (исключающее «или») определяется таблицей истинности 4.2.1.

Таблица 4.2.1. Таблица истинности

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0

Другими словами, исключающее «или» двух булевских значений равно единице, если ровно одно из них равно единице.

Побитовая операция исключающего «или» для двух чисел определяется над их двоичными представлениями. Заданная операция применяется к каждой паре соответствующих битов обоих чисел.

Например, $100 \text{ xor } 50$ можно вычислить следующим образом: $100 = (01100100)_2$, $50 = (00110010)_2$.

$$\begin{array}{r}
 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \\
 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \\
 \hline
 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0
 \end{array}$$

В итоге $(01010110)_2 = 86$.

Отметим, что в языках C++ и Python операция побитового исключающего «или» для двух переменных x и y записывается при помощи символа крышки $x \wedge y$.

В качестве примера рассмотрим программу, которая получает число 589827 из заданного числа 5. В столбцах **dec** и **bin** (таблица 4.2.2) записано десятичное и двоичное представление числа, полученного в результате исполнения команды.

Таблица 4.2.2

num	com	dec	bin
0		5	00000000000000000101
1	r 0 1	10	00000000000000001010
2	x 0 1	15	00000000000000001111
3	r 2 3	120	00000000000001111000
4	x 1 3	114	00000000000001110010
5	r 4 15	589827	10010000000000000011

Преподавателю понравился компьютер Алисы, но он попросил ее продемонстрировать его возможности. Для этого она должна написать n **различных** программ,

чтобы получить заданные числа $a_1, \dots, a_n$ из одного и того же заданного числа x . Преподаватель дал достаточно много чисел, и Алиса не успевает написать для них программы. Напишите программу, которая будет автоматически генерировать требуемые программы. Обратите внимание: в задаче не требуется находить кратчайшую программу. Необходимо лишь позаботиться о том, чтобы количество команд в ней не превосходило 127.

Формат входных данных

В первой строке на вход поступает единственное натуральное число x — исходное число для программ Алисы, $1 \leq x \leq 2^{20} - 1$.

Во второй строке на вход поступает единственное натуральное число n — количество чисел для получения которых требуется написать программы, $1 \leq n \leq 2000$.

В третьей строке на вход поступают числа $a_1, \dots, a_n$, для получения которых требуется написать программу, $0 \leq a_i \leq 2^{20} - 1$.

Формат выходных данных

Для каждого из чисел во входных данных требуется вывести полученные программы. В первой строке вывода требуется указать одно число s — количество команд в программе, $-1 \leq s \leq 127$. Если его невозможно получить за 127 или менее команд, то s должно быть равно (-1) . Далее в s строках требуется записать команды программы в указанном формате без порядкового номера команды. Требуемое число должно быть получено последней командой программы.

Примеры

Пример №1

Стандартный ввод
5
4
0 1 5 589827
Стандартный вывод
1
x 0 0
-1
0
5
r 0 1
x 0 1
r 2 3
x 1 3
r 4 15

Примечания

Для первого числа одна команда $x \leftarrow 0$ дает ноль, так как a^a равно нулю для любого a .

Для второго числа можно доказать, что получить 1 невозможно.

Третье число совпадает с исходным. Можно ничего не делать.

Программа для четвертого числа разобрана в примере.

Решение

Обозначим операцию циклического сдвига числа x на k бит за $x \triangleleft k$.

Очевидно имеют место тождества

$$(x \oplus y) \triangleleft k = (x \triangleleft k) \oplus (y \triangleleft k),$$

$$(x \triangleleft a) \triangleleft b = x \triangleleft (a + b).$$

Тогда, используя эти тождества, можно преобразовать любую формулу к виду

$$(x \triangleleft k_1) \oplus (x \triangleleft k_2) \oplus \dots \oplus (x \triangleleft k_m).$$

Теперь можно перебрать все формулы такого вида и для каждой формулы найти число, которое получается в результате ее вычисления. В прилагаемой программе в списке r хранятся все циклические сдвиги x , а каждая формула указанного вида кодируется в виде числа, двоичная запись которого содержит единицы на позициях $k_1, \dots, k_m$. В списке x для каждого возможного двадцатизначного двоичного числа хранится код его формулы или (-1) , если это число невозможно получить.

Вторая часть программы выводит ответ в виде формулы указанного вида.

Пример программы-решения

Ниже представлено решение на языке Python.

Python

```

1  bits = 20
2  r = [0]*bits
3  r[0] = int(input())
4  for i in range(bits-1):
5      r[i+1] = ((r[i]>>(bits-1)) + (r[i]<<1)) & (2**bits-1)
6  x = [-1]*(2**bits)
7  for p in range(2**bits):
8      t = 0
9      for k in range(bits):
10         if ((p>>k)&1)==1:
11             t^=r[k]
12     x[t]=p
13 n = int(input())
14 for p in map(int, input().split()):
15     if x[p]==-1:
16         print(-1)

```

```

17     else:
18         print(bits+bin(x[p]).count('1'))
19         for i in range(bits-1):
20             print('r', i, 1)
21         print('x', 0, 0)
22         n=bits
23         for k in range(bits):
24             if ((x[p]>>k)&1)==1:
25                 print('x', k, n)
26                 n+=1

```

Критерии оценивания

Тест с номером 1 совпадает с тестом из условия задачи. Баллы за него не начисляются.

Успешное прохождение каждого теста с номерами 2–31 оценивается в 1 балл.

В тестах с номерами 2–3 выполняется $x = 1$.

В тестах с номерами 4–7 выполняется $x = 3$.

В тестах с номерами 8–11 выполняется $x = 7$.

В тестах с номерами 12–19 выполняется $n \leq 20$.

Задача 4.2.1.5. Алиса и алгоритм (30 баллов)

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

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

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Алиса пишет программу по заданному алгоритму. На вход алгоритму подается некоторое целое число x и последовательность $a_1, \dots, a_n$. Алгоритм является циклическим. На каждом шаге цикла для i от 1 до n переменная x сравнивается с a_i и, если $x \geq a_i$, то $x = x - a_i$. Полученное после завершения цикла значение переменной x является ответом. Ниже записан текст этой программы в виде функции на языке Python.

Python

```

1 def algo(x, A):
2     for a in A:
3         if x >= a:
4             x -= a
5     return x==0

```

Алисе задалась вопросом, сколько существует значений x , при которых алгоритм для фиксированной последовательности $a_1, \dots, a_n$ дает ответ 0. Ей кажется, что для

этого придется перебрать все возможные значения x от 0 до $\sum_{i=1}^n a_i$, но она понимает, что такая программа будет работать слишком долго.

Сможете ли вы написать программу, которая решит задачу, поставленную Алисой, быстрее?

Программа должна ответить на t независимых запросов во входных данных.

Формат входных данных

В первой строке на вход подается единственное натуральное число t — количество наборов входных данных ($1 \leq t \leq 100$).

Далее записаны сами наборы входных данных. Каждый набор входных данных состоит из двух строк. В первой из них записано одно натуральное число n — количество элементов в последовательности ($1 \leq n \leq 200\,000$). Вторая строка набора содержит последовательность натуральных чисел $a_1, \dots, a_n$ ($1 \leq a_i \leq 10^6$).

Гарантируется, что сумма n по всем наборам входных данных не превышает 200 000. **Также гарантируется, что сумма всех значений a_i во входном файле не превышает $2 \cdot 10^8$.**

Формат выходных данных

Для каждого набора входных данных выведите ответ в виде одного целого числа. Числа можно выводить как в одной строке через пробел, так и в разных.

Примеры

Пример №1

Стандартный ввод
3
5
1 2 3 4 5
5
5 4 3 2 1
4
5 1 5 1
Стандартный вывод
6 16 8

Примечания

В первом тестовом случае ответ 0 может быть получен при начальных значениях x , равных 0, 1, 3, 6, 10, 15.

Во втором тестовом случае ответ 0 может быть получен при всех начальных значениях x от 0 до 15.

В третьем тестовом случае ответ 0 может быть получен при начальных значениях x , равных 0, 1, 2, 5, 6, 7, 11, 12.

Решение

Рассмотрим решение этой задачи методом динамического программирования. Обозначим за $dp(k, x)$ количество чисел, из которых можно получить число x , выполнив данную программу для первых k чисел заданного списка. Очевидно $dp(0, x) = 1$. Проанализировав программу, можно прийти к формуле

$$dp(k, x) = \begin{cases} dp(k-1, x) + dp(k-1, x + a_k), & x < a_k, \\ dp(k-1, x + a_k), & x \geq a_k. \end{cases}$$

Действительно, если после очередного шага было получено число $x < a_k$, то на этом шаге вычитание могло выполняться или не выполняться. В противном случае вычитание производилось наверняка. Ответом будет $dp(n, 0)$.

Отметим, что если эту формулу записать в виде программы без преобразований, то количество сложений будет равно $\sum_{i=1}^n a_i$. С учетом ограничений на сумму такое количество сложений может быть выполнено за отведенное время. Таким образом, надо найти реализацию, при которой данные не будут перемещаться по памяти. Также заметим, что если $s \geq \max a_i$, то $dp(k, s) = 1$. Это означает, что для всех s , больших 10^6 , значением $dp(k, s)$ будет единица. Таким образом, достаточно хранить 10^6 значений, а единицы добавлять при необходимости.

Рассмотрим преобразование данных на одном шаге.

$$\begin{array}{cccccccccccccccc} d_0 & d_1 & \cdots & d_{a-1} & d_a & d_{a+1} & \cdots & d_{r-a} & d_{r-a+1} & \cdots & d_r & 1 & \cdots \\ d_0 + d_a & d_1 + d_{a+1} & \cdots & d_{a-1} + d_{2a-1} & d_{2a} & d_{2a+1} & \cdots & d_r & 1 & \cdots & 1 & 1 & \cdots \end{array}$$

Можно заметить, что если первые a элементов добавить к следующим a элементам, а потом удалить, а также добавить a единиц в конец последовательности, то получим требуемое преобразование. Для реализации можно использовать кольцевой буфер требуемого размера.

Пример программы-решения

Ниже представлено решение на языке Python.

Python

```
1 maxval = 1000002
2 t = int(input())
3 for _ in range(t):
4     n = int(input())
5     rng = [1]*maxval
6     pos = 0
7     for a in map(int, input().split()):
8         for i in range(a):
9             rng[(pos+a)%maxval] += rng[pos%maxval]
10            rng[pos%maxval] = 1
```

```
11         pos+=1
12     print(rng[pos%maxval],end=' ')
```

Критерии оценивания

Тест с номером 1 совпадает с тестом из условия задачи. Баллы за него не начисляются.

Успешное прохождение каждого теста с номерами 2–31 оценивается в 1 балл.

В тестах с номерами 2–9 сумма n по всем тестовым случаям не превосходит 1 000, при этом выполняется $a_i \leq 1\,000$ и сумма всех a_i по всем тестовым случаям не превосходит 50 000.

В тестах с номерами 10–15 сумма n по всем тестовым случаям не превосходит 20 000, при этом выполняется $a_i \leq 1\,000$.

В тестах с номерами 16–21 сумма n по всем тестовым случаям не превосходит 100 000, при этом выполняется $a_i \leq 1\,000$.

4.2.2. Математика. 8–9 классы

Задача 4.2.2.1. (10 баллов)

Тема: разложение натурального числа на множители.

Условие

Найдите какое-нибудь решение уравнения $n \cdot S(n) = 2025$, где n — это некоторое натуральное число и $S(n)$ — сумма его цифр.

Решение

Раскладывая число 2025 на два сомножителя, можем найти, что подходит, например, случай $2025 = 9 \cdot 225$ и $2 + 2 + 5 = 9$.

Ответ: 225.

Критерии оценивания

Найдено какое-либо решение уравнения — 10 баллов.

Задача 4.2.2.2. (20 баллов)

Тема: текстовая задача.

Условие

Кибернетический Дед Мороз с помощью генератора случайных чисел раздал 42 конфеты десяти программистам. Докажите, что найдутся два программиста, которые получили одинаковое число конфет или не получили конфет вообще.

Решение

Предположим, что все программисты получили разное число конфет, тогда их наименьшее количество равно $0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$ конфет, что противоречит условию. Значит, найдутся программисты, получившие одинаковое число конфет.

Критерии оценивания

Доказано, что минимальная сумма конфет равна 45 — 12 баллов.

Задача 4.2.2.3. (20 баллов)

Тема: элементы теории чисел.

Условие

Сколько существует натуральных n таких, что числа n и $n + 2024$ имеют нечетное число делителей?

Решение

Заметим, что если у натурального числа нечетное число делителей, то данное число является квадратом некоторого числа.

Пусть $n = a^2$, $n + 2024 = b^2$. Распишем их разность:

$$b^2 - a^2 = (b - a)(b + a) = 2024.$$

Заметим, что числа $b - a$ и $b + a$ имеют одинаковую четность и натуральные, тогда это два натуральных четных числа (нечетными быть не могут, так как их произведение четно).

Разложим правую часть равенства на простые множители:

$$2024 = 2 \cdot 2 \cdot 2 \cdot 11 \cdot 23.$$

Так как в левой части равенства стоит два четных множителя, то эти числа нужно разбить на две группы так, чтобы произведение чисел в каждой группе было четным. Разместим сначала в каждую группу по одной двойке, и тогда нужно разбить произведение $2 \cdot 11 \cdot 23$ на два множителя, причем больший множитель — это $(a + b)$, а меньший — $(a - b)$. Сделать это можно четырьмя способами:

- 1 и $2 \cdot 11 \cdot 23$;
- 2 и $11 \cdot 23$;
- 11 и $2 \cdot 23$;
- $2 \cdot 11$ и 23.

В каждом способе однозначно определяются a и b , а значит, и однозначно определяем n .

Ответ: 4.

Критерии оценивания

Показано, что оба числа являются квадратами — 3 балла.

Расписана формула разности квадратов — 3 балла (суммируется с предыдущим).

Замечено, что $a + b$ и $a - b$ — одной четности — 4 балла (суммируется с предыдущими).

Найдены все возможные случаи значений $a + b$ и $a - b$ — 6 баллов.

Сказано (или найдено), что в каждом из четырех случаев однозначно определяются числа a и b — 4 балла.

Задача 4.2.2.4. (25 баллов)

Темы: алгебра, доказательство неравенств.

Условие

Найдите наибольшее значение выражения:

$$x^2 + 8x^2y^2 + y^2,$$

где x и y — неотрицательные числа, сумма которых не превосходит 1.

Решение

Вариант 1. По условию задачи $0 \leq x + y \leq 1$. Тогда $(x + y)^2 \leq 1$.

С другой стороны, заметим, что $4xy \leq (x + y)^2$. Отсюда следует, что $4xy \leq 1$. Так как числа x и y неотрицательные, то умножим левую и правую часть последнего неравенства на $2xy$ и получим, что $8x^2y^2 \leq 2xy$. Оценим требуемое выражение:

$$x^2 + 8x^2y^2 + y^2 \leq x^2 + 2xy + y^2 \leq (x + y)^2 \leq 1.$$

Таким образом, максимальное значение выражения равно 1 и достигается, например, при $x = 1, y = 0$.

Вариант 2. Рассмотрим свойства функции $f(x, y) = x^2 + 8x^2y^2 + y^2$. Она является монотонно возрастающей по x и по y при $x, y \geq 0$. Это значит, что максимальное значение достигается там, где $x + y = 1$. Выразив $y = 1 - x$ и подставив в функцию $f(x, y)$, получим функцию:

$$\phi(x) = x^2 + 8x^2(1 - x)^2 + (1 - x)^2.$$

Заметим, что $\phi(x) = \phi(1 - x)$, то есть функция $\phi(x)$ симметрична. Это означает, что ее можно упростить, сделав замену $z = x - \frac{1}{2}$:

$$\left(z + \frac{1}{2}\right)^2 + 8\left(z + \frac{1}{2}\right)^2 \left(z - \frac{1}{2}\right)^2 + \left(z - \frac{1}{2}\right)^2 = 8z^4 - 2z^2 + 1 = g(z), z^2 \in \left[0, \frac{1}{4}\right].$$

Эта функция является биквадратным трехчленом, его вершина в точке $z^2 = -\frac{-2}{2 \cdot 8} = \frac{1}{8}$. В этой точке достигается его минимум, а максимум достигается на концах отрезка: $g_{\max} = g(0) = g\left(\frac{1}{4}\right) = 1$.

Ответ: 1.

Критерии оценивания**Критерии оценивания для решения 1:**

Сделана оценка $(x + y)^2 \leq 1$ — 2 балла.

Сделана оценка $4xy \leq (x + y)^2$ — 3 балла.

Получено неравенство $8x^2y^2 \leq 2xy$ — 8 баллов.

Получено неравенство $x^2 + 8x^2y^2 + y^2 \leq 1$ — 8 баллов.

Получено требуемое наибольшее значение и указаны переменные, при которых оно достигается — 4 балла.

Критерии оценивания для решения 2:

Использована монотонность функции $f(x, y)$ или другой корректный способ доказательства, чтобы приравнять в точке максимума $x + y = 1$ — 2 балла.

Если данный пункт используется без использования монотонности функции или других соображений, то 1 балл.

Верно получен вид для функции $\phi(x)$ — 3 балла.

Сделана замена $z = x - \frac{1}{2}$ или другой способ преобразования, позволяющий упростить функцию до вида, позволяющего найти максимум — 8 баллов.

Получена функция $g(z) = 8z^4 - 2z^2 + 1$ или какая-то другая, которая позволяет получить правильное решение — 8 баллов.

Получено требуемое наибольшее значение и указаны переменные (можно только те, что после замены), при которых оно достигается — 4 балла.

Задача 4.2.2.5. (25 баллов)

Тема: планиметрия.

Условие

В невыпуклом четырехугольнике $ABCD$ углы A , B и D равны по 45° . Докажите, что середины его сторон являются вершинами квадрата.

Решение

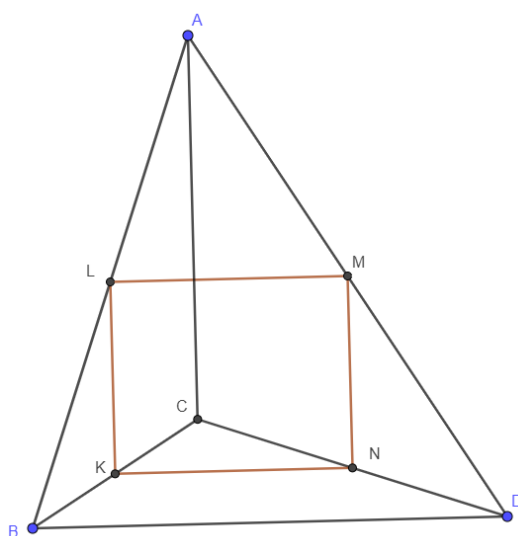


Рис. 4.2.2

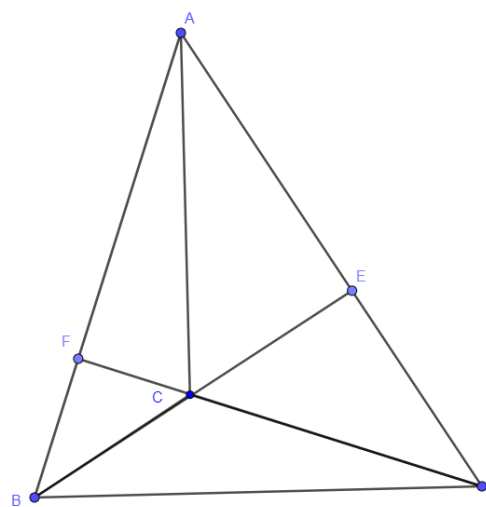


Рис. 4.2.3

Обозначим середины сторон AB , BC , CD и DA через L , K , N , M соответственно (рис. 4.2.2). Заметим, что каждая из сторон четырехугольника $LKNM$ является средней линией в некотором треугольнике и выполняются условия: $LM = KN = \frac{1}{2}BD$, $LK = MN = \frac{1}{2}AC$. Тогда чтобы доказать, что $LKNM$ — квадрат, достаточно показать равенство и перпендикулярность BD и AC .

Продлим BC и DC до противоположных сторон и точки пересечения обозначим соответственно через E и F (рис. 4.2.3). Заметим, что так как по условию $\angle ABE = \angle BAE = 45^\circ$, то $\angle AEB = 90^\circ$, то есть $BE \perp AD$. Аналогично, так как $\angle ADF = \angle DAF = 45^\circ$, то $\angle AFD = 90^\circ$, то есть $DF \perp AB$. Таким образом, получаем, что C — это точка пересечения высот (ортоцентр) в треугольнике ABD . И тогда по свойству ортоцентра $AC \perp BD$.

Заметим, что треугольники AEC и BED равны по первому признаку ($AE = BE$ из равнобедренного прямоугольного треугольника AEB и $CE = DE$ из равнобедренного прямоугольного треугольника CED , а $\angle BED = \angle AEC = 90^\circ$). Следовательно, $AC = BD$ как соответствующие элементы. Таким образом, получаем, что $AC \perp BD$ и $AC = BD$. Что и требовалось доказать.

Критерии оценивания

Доказано, что стороны четырехугольника $LMNK$ являются средними линиями в соответствующих треугольниках — 5 баллов.

Из пункта выше сделано замечание, что $LMNK$ будет являться квадратом, если отрезки AC и BD будут равны, и прямые, содержащие их, перпендикулярны, но дальнейших продвижений нет — 3 балла.

Сделано дополнительное построение точек E и F , но дальнейших продвижений нет — 1 балл.

Доказано, что C — точка пересечения высот треугольника ABD — 5 баллов.

Доказано, что $AC \perp BD$ — 3 балла.

Доказано равенство $AC = BD$ — 7 баллов.

Сделан окончательный вывод, содержащий все предыдущие пункты — 1 балл.

Все баллы пунктов суммируются.

4.2.3. Математика. 10–11 классы

Задача 4.2.3.1. (10 баллов)

Тема: функция.

Условие

Приведите пример функции, отличной от константы, для которой справедливо:

$$f(3) \cdot f(5) = 2025, f(3) \neq f(5).$$

Решение

$2025 = 3^4 \cdot 5^2 = 3^{7-3} \cdot 5^{7-5}$. Для функции $f(x) = x^{7-x}$ выполняются указанные условия.

Замечание: существуют и другие функции, например, $f(x) = 3x^2$.

Ответ: например, $f(x) = x^{7-x}$ или $f(x) = 3x^2$.

Критерии оценивания

Приведен верный пример любой функции — 10 баллов.

Задача 4.2.3.2. (20 баллов)

Тема: тождественные преобразования.

Условие

Найдите сумму:

$$\frac{3}{1! + 2! + 3!} + \frac{4}{2! + 3! + 4!} + \cdots + \frac{2025}{2023! + 2024! + 2025!}.$$

Напомним, что число $n! = 1 \cdot 2 \cdot \dots \cdot n$ называется факториалом натурального числа n .

Решение

Заметим, что каждая дробь имеет вид:

$$\frac{n}{(n-2)! + (n-1)! + n!}.$$

Преобразуем данную дробь:

$$\begin{aligned} \frac{n}{(n-2)! + (n-1)! + n!} &= \frac{n}{(n-2)!(1 + n - 1 + n^2 - n)} = \\ &= \frac{n}{(n-2)!n^2} = \frac{1}{(n-2)!n} = \frac{1}{(n-1)!} - \frac{1}{n!}. \end{aligned}$$

Заменим в исходном выражении каждую дробь по данному правилу:

$$\begin{aligned} \frac{3}{1! + 2! + 3!} + \frac{4}{2! + 3! + 4!} + \cdots + \frac{2025}{2023! + 2024! + 2025!} &= \\ = \left(\frac{1}{2!} - \frac{1}{3!} \right) + \left(\frac{1}{3!} - \frac{1}{4!} \right) + \cdots + \left(\frac{1}{2024!} - \frac{1}{2025!} \right) &= \frac{1}{2!} - \frac{1}{2025!}. \end{aligned}$$

Ответ: $\frac{1}{2!} - \frac{1}{2025!}$.

Критерии оценивания

Доказано, что каждая из дробей представима в виде $\frac{1}{(n-1)!} - \frac{1}{n!}$, но дальнейших продвижений нет — 10 баллов.

Задача 4.2.3.3. (20 баллов)

Тема: элементы теории чисел.

Условие

Сколько существует натуральных n таких, что числа n и $n + 2024$ имеют нечетное число делителей?

Решение

Заметим, что если у натурального числа нечетное число делителей, то данное число является квадратом некоторого числа.

Пусть $n = a^2$, $n + 2024 = b^2$. Распишем их разность:

$$b^2 - a^2 = (b - a)(b + a) = 2024.$$

Заметим, что числа $b - a$ и $b + a$ имеют одинаковую четность и натуральные, тогда это два натуральных четных числа (нечетными быть не могут, так как их произведение четно).

Разложим правую часть равенства на простые множители:

$$2024 = 2 \cdot 2 \cdot 2 \cdot 11 \cdot 23.$$

Так как в левой части равенства стоит два четных множителя, то эти числа нужно разбить на две группы так, чтобы произведение чисел в каждой группе было четным. Разместим сначала в каждую группу по одной двойке, и тогда нужно разбить произведение $2 \cdot 11 \cdot 23$ на два множителя, причем больший множитель — это $(a + b)$, а меньший — $(a - b)$. Сделать это можно четырьмя способами:

- 1 и $2 \cdot 11 \cdot 23$;
- 2 и $11 \cdot 23$;
- 11 и $2 \cdot 23$;
- $2 \cdot 11$ и 23.

В каждом случае однозначно определяются a и b , а значит, и однозначно определяем n .

Ответ: 4.

Критерии оценивания

Показано, что оба числа являются квадратами — 3 балла.

Расписана формула разности квадратов — 3 балла (суммируется с предыдущим).

Замечено, что $a + b$ и $a - b$ — одной четности — 4 балла (суммируется с предыдущими).

Найдены все возможные случаи значений $a + b$ и $a - b$ — 6 баллов.

Сказано (или найдено), что в каждом из четырех случаев однозначно определяются числа a и b — 4 балла.

Задача 4.2.3.4. (25 баллов)

Тема: планиметрия.

Условие

Из точки A к окружности проведена касательная AB и секущая AC , которая пересекает окружность в точке D так, что $AD : DC = 4 : 3$. Найдите отношение расстояния от центра окружности до хорды BC к ее радиусу, если известно, что площадь треугольника ABC равна $42\sqrt{3}$ и $\cos \angle BAC = \frac{\sqrt{7}}{14}$.

Решение

Пусть точка O — центр рассматриваемой окружности, OE — расстояние от точки O до хорды BC и OB — радиус окружности. Тогда требуемое отношение из прямоугольного треугольника OEB будет равно $\frac{OE}{OB} = \sin \angle OBE$.

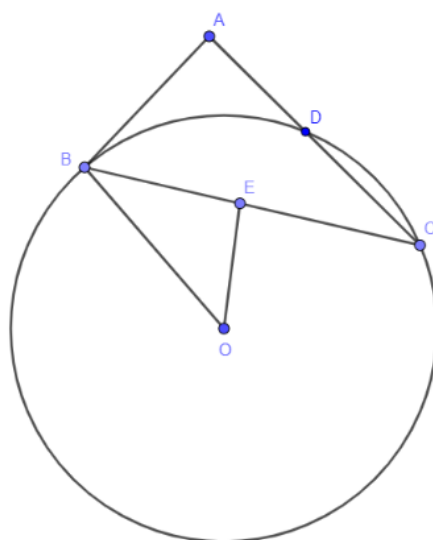


Рис. 4.2.4

Из точки A к окружности проведена касательная AB и секущая AC , причем $AD : DC = 4 : 3$. Обозначим: $AD = 4x$, $AC = 7x$. Тогда по теореме о касательной и секущей имеем:

$$AB^2 = AD \cdot AC = 4x \cdot 7x = 28x^2, \text{ т. е. } AB = 2\sqrt{7}x.$$

Запишем площадь треугольника ABC :

$$S = \frac{1}{2} AB \cdot AC \cdot \sin \angle BAC = \frac{1}{2} \cdot 2\sqrt{7}x \cdot 7x \cdot \sin \angle BAC = 7\sqrt{7}x^2 \cdot \sin \angle BAC.$$

Из основного тригонометрического тождества найдем $\sin \angle BAC$:

$$\sin \angle BAC = \sqrt{1 - \cos^2 \angle BAC} = \sqrt{1 - \left(\frac{\sqrt{7}}{14}\right)^2} = \frac{3\sqrt{21}}{14}.$$

Зная, что по условию задачи площадь треугольника ABC равна $42\sqrt{3}$, получим уравнение:

$$42\sqrt{3} = 7\sqrt{7}x^2 \cdot \frac{3\sqrt{21}}{14}.$$

Решая данное уравнение, найдем $x = 2$.

Таким образом, треугольник ABC имеет стороны $AB = 4\sqrt{7}$ и $AC = 14$. По теореме косинусов найдем его третью сторону:

$$BC^2 = AB^2 + AC^2 - 2 \cdot AB \cdot AC \cdot \cos \angle BAC = (4\sqrt{7})^2 + 14^2 - 2 \cdot 4\sqrt{7} \cdot 14 \cdot \frac{\sqrt{7}}{14} = 252, \\ BC = 6\sqrt{7}.$$

Запишем теорему синусов для треугольника ABC :

$$\frac{AC}{\sin \angle ABC} = \frac{BC}{\sin \angle BAC}.$$

Найдем из этого равенства $\sin \angle ABC$:

$$\sin \angle ABC = \frac{AC}{BC} \cdot \sin \angle BAC = \frac{14}{6\sqrt{7}} \cdot \frac{3\sqrt{21}}{14} = \frac{\sqrt{3}}{2}.$$

Таким образом, получаем, что $\angle ABC = 60^\circ$.

Тогда так как AB — касательная к окружности, то:

$$\angle OBE = \angle ABO - \angle ABC = 30^\circ \text{ и } \frac{OE}{OB} = \sin 30^\circ = 0,5.$$

Ответ: 0,5.

Критерии оценивания

Найдено соотношение между сторонами AB и AC , но дальнейших продвижений нет — 2 балла.

Найдены длины сторон AB и AC — 8 баллов.

Найдена длина стороны BC — 5 баллов.

Найден $\sin \angle ABC$ — 5 баллов.

Найден $\angle OBE$ — 2 балла.

Найдено требуемое отношение — 3 балла.

Все баллы суммируются.

Задача 4.2.3.5. (25 баллов)*Тема: тригонометрическое уравнение.***Условие**

Решите уравнение:

$$16 \sin^4 \left(\frac{x}{3} \right) + 1 - 4 \sin^2 \left(\frac{x}{3} \right) \cos(4x) = 4 \sin^2 \left(\frac{x}{3} \right) \cos^2(4x).$$

РешениеСделав замену $t = \sin^2 \left(\frac{x}{3} \right)$, получим квадратное уравнение относительно t :

$$16t^2 - 4t(\cos^2(4x) + \cos(4x)) + 1 = 0$$

Дискриминант этого уравнения будет неположительным в силу ограниченности функции $y = \cos x$. Действительно:

$$D = 16(\cos^2(4x) + \cos(4x))^2 - 64 \leq 0.$$

Значит, единственное решение для t получается при $\cos(4x) = 1$. При этом $t = \frac{1}{4}$, то есть $\sin \left(\frac{x}{3} \right) = \pm \frac{1}{2}$.

Таким образом, получаем первую систему уравнений:

$$\begin{cases} \cos(4x) = 1, \\ \sin \left(\frac{x}{3} \right) = \frac{1}{2} \end{cases}$$

и

$$\begin{cases} \cos(4x) = 1, \\ \sin \left(\frac{x}{3} \right) = -\frac{1}{2}. \end{cases}$$

Решая данные системы уравнений, получим четыре серии решений:

$$x_{1,2} = \pm \frac{\pi}{2} + 6\pi n, \quad x_{3,4} = \pm \frac{5\pi}{2} + 6\pi k, \quad k, n \in \mathbb{Z}.$$

Ответ: $x_{1,2} = \pm \frac{\pi}{2} + 6\pi n, \quad x_{3,4} = \pm \frac{5\pi}{2} + 6\pi k, \quad k, n \in \mathbb{Z}.$ **Критерии оценивания**Сделана замена $t = \sin^2 \left(\frac{x}{3} \right)$, но дальнейших продвижений нет — 2 балла.Доказано, что уравнение относительно t имеет единственное решение при $\cos(4x) = 1$ — 10 баллов.

Получены системы уравнений — 3 балла.

Найдены решения систем уравнений — 10 баллов.

Примечание. Решение задачи также может быть получено, если сделать замену $t = \cos^2(4x)$. В этом случае все критерии оценивания являются схожими.

4.3. Инженерный тур

4.3.1. Общая информация

Беспроводная связь стала неотъемлемой частью современных технологий, которые прочно вошли в повседневную жизнь, промышленную и научную сферы. Чтобы информация надежно передавалась на расстоянии, существует множество различных решений, начиная от протоколов и помехоустойчивых кодов, заканчивая слежением за подвижным источником сигнала (например, спутником). Создание протоколов эффективной и надежной передачи данных — фундаментальная проблема связи, решение которой зависит от конкретных условий среды и канала.

На профиле команды столкнутся с комплексом заданий по организации беспроводных каналов связи в разных средах. Участники смогут буквально увидеть и услышать сигнал, поработать с передачей данных и создать свои помехозащищенные протоколы.

4.3.2. Легенда задачи

Беспроводная связь, столь удобная и привычная, состоит из набора технологий, решающих частные задачи, общая цель которых — надежно и быстро передать данные на расстоянии.

Участникам предлагается решить комплекс задач по организации беспроводных каналов связи в различных средах и условиях. Они выполняются на стендах, представляющих настоящий беспроводной канал со своими особенностями и ограничениями. Каждая задача затрагивает свой аспект беспроводной передачи информации — от устойчивой к потерям упаковки данных до настройки слежения за подвижным источником сигнала и борьбы с шумами.

Технологии передачи больших зашифрованных массивов данных, слежение за подвижным источником сигнала применяется в отраслях, где нужна связь с автономным устройством в труднодоступном месте, куда нельзя физически добраться: при ремонте автоматических космических аппаратов, работе с подводными роботами и т. п. У участников есть возможность не просто изучить эти технологии, но даже создать и протестировать собственные протоколы связи.

Навык разработки собственных протоколов — базовый в технологиях связи, огромном растущем рынке в мире с увеличивающимся спросом на специалистов. Дальнейшее развитие беспроводных технологий связи — это не только технологии связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов и других объектов интернета вещей — все они требуют создания протоколов.

В настоящее время количество различных протоколов связи, количество систем, где их важно проектировать непрерывно растет, но понимание базовых принципов и способность конструировать такие каналы падает из-за усложнения протоколов. Поэтому важно работать, с одной стороны, с максимально современной постановкой

задачи, а с другой — с максимальной прозрачностью и возможностью ее увидеть и «пощупать».

4.3.3. Требования к команде и компетенциям участников

Количество участников в команде: 4–5 человек.

Компетенции, которыми должны обладать члены команды:

1. **Программист** — программирование адаптивной системы слежения; работа с протоколами связи.
2. **Аналитик-исследователь** — анализ характеристик шума в каналах; выбор схемы помехоустойчивого кодирования; помехоустойчивое кодирование для оптического канала.
3. **Аналитик-исследователь** — работа с данными и основами корреляционного анализа, с механикой стенда; расшифровка кодов и характеристик стенда.
4. **Программист** — обработка сигналов; работа с данными и владение основами корреляционного анализа.
5. **Капитан** — сбор и анализ решений подзадач от всех специалистов, описанных выше; обсуждение и коррекция полученных результатов для более эффективного решения общей задачи; организация командной работы; капитан должен обладать знаниями по всем основным темам, уметь разбираться в коде.

4.3.4. Оборудование и программное обеспечение

На инженерном туре команды профиля Технологии беспроводной связи работают со стендами:

- «Узконаправленные низкоэнергетические каналы связи» — УНКС (см. рис. 4.3.1);
- «Каналы связи и кодирование» — О-БТС (см. рис. 4.3.2);
- «Турнир Юных Киберфизиков «Акустика» » — ТЮК-А (см. рис. 4.3.3);
- «Акустический полигон» — А-Полигон (рис. 4.3.4, 4.3.5).

Стенды УНКС и О-БТС входят в образовательный комплекс «Беспроводные технологии связи», разработанный компанией ИнСитиЛаб. Они созданы в логике моделирования реальных инженерных задач по созданию протоколов связи. Программно-аппаратная платформа стенда УНКС позволяет работать с адаптивными алгоритмами и точностью передачи сигнала. При создании стенда разработчиками и экспертами выступали специалисты ИСЗФ СО РАН с реальным опытом работы проектирования систем слежения — современных радаров.



Рис. 4.3.1. Стенд «Узконаправленные низкоэнергетические каналы связи» (УНКС)

Стенд О-БТС стал нововведением в этом году, добавив задачи на оптический канал связи.



Рис. 4.3.2. Стенд «Каналы связи и кодирование» (О-БТС)

При работе с комплектом для проведения турнира юных киберфизиков «Акустика-1», разработанном компанией «ИнСитиЛаб» в рамках Национальной киберфизической платформы, участники взаимодействуют с акустическим каналом связи.

ТЮК-А позволяет наглядно коммуницировать с аналоговыми и цифровыми сигналами в физическом акустическом канале связи, с разными типами модуляции и помехоустойчивым кодированием.

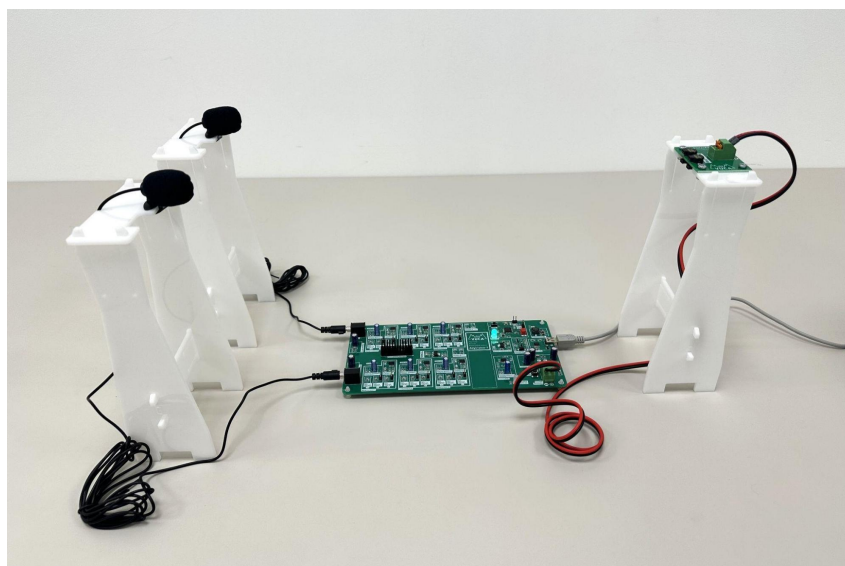


Рис. 4.3.3. Комплекс «Турнир Юных Киберфизиков «Акустика» » (ТЮК-А)

Еще одним нововведением этого года стал комплекс «Акустический полигон» — модификация ТЮК-А с шумоизолированным боксом. Стенд позволяет ставить комплексные задачи на акустическом канале, на которые не повлияет окружающий шум.



Рис. 4.3.4. Комплекс «Акустический полигон» (А-Полигон)

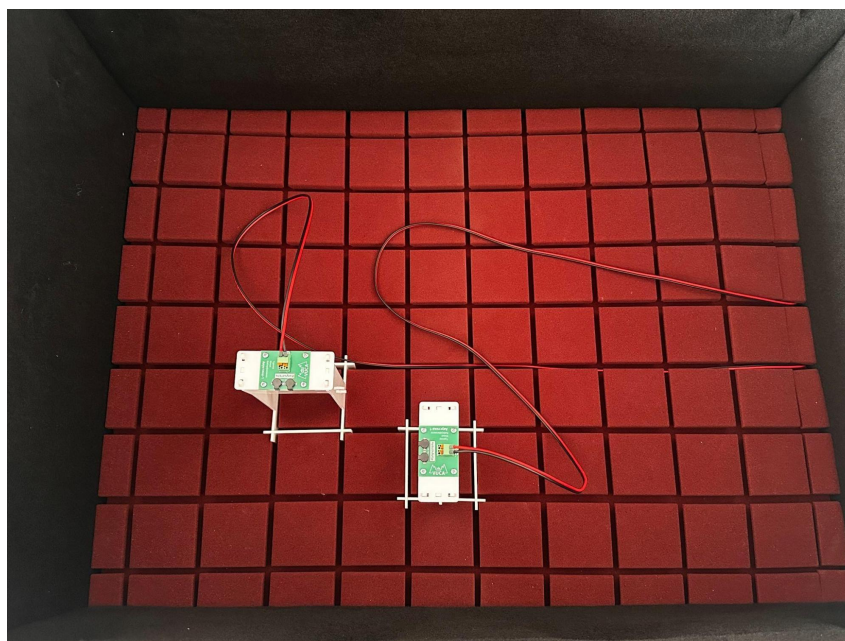


Рис. 4.3.5. Комплекс «Акустический полигон» (А-Полигон внутри)

На время инженерного тура каждая команда получает по одному комплекту «ТЮК-Акустика» и стенду «Каналы связи и кодирование». Взаимодействие с ними осуществляется на двух компьютерах, предоставленных каждой команде, с предустановленным ПО для соответствующих стендов.

Два стенда «Узконаправленные низкоэнергетические каналы связи» (УНКС) размещены в общем доступе на площадке проведения. Участники эксплуатируют их в порядке живой очереди с ограничением времени сеанса. Взаимодействие со стендом выполняется через рабочую станцию с предустановленным ПО комплекса «Беспроводные технологии связи» с использованием только тех библиотек, которые были предустановлены в среде.

Каждая команда тренировочно работает со своим комплексом. Для обеспечения равенства условий всех команд проверка решений производится на судейских контрольных стендах (одна задача — один стенд).

Таблица 4.3.1

Наименование	Описание
Стенд «Узконаправленные низкоэнергетические каналы связи» образовательного комплекса «Беспроводные технологии связи» — УНКС	Позволяет моделировать реальную ситуацию для систем связи нового типа между автономными роевыми объектами. Задачи на стенде посвящены установлению слежения движущихся объектов друг за другом и адаптивной дешифрации в реальном времени. Во время работы участники создают свои протоколы передачи данных под конкретные условия шумов, требования к надежности, характеру помех в канале, скорости передачи.

Стенд «Каналы связи и кодирование» образовательного комплекса «Беспроводные технологии связи» — О-БТС	Предназначен для обучения работе с аналоговыми и цифровыми сигналами, низкоуровневыми протоколами, современными системами кодирования. На стенде моделируется оптический видимый канал связи с виртуальной и физической модуляцией (яркость света и вращающийся диск), регулируемой пропускаемостью сигнала. Со стендом на примере оптического сигнала можно эффективнее освоить теорию сигналов, двоичные коды, кодирование и декодирование, анализ данных, анализ кода.
Комплект «Турнир Юных Киберфизиков “Акустика”» — ТЮК-А	Представляет собой акустический канал связи с передатчиком (излучателем) и двумя микрофонами (приемниками), размещенными на расстоянии друг от друга. Параметры усилителей микрофонов можно настраивать, замыкая участки цепи. На плату передается звуковой файл, воспроизводимый излучателем. Звук с микрофонов передается обратно и записывается в виде двух принятых сигналов как набор измерений АЦП. Со стендом на примере акустического сигнала можно эффективнее освоить теорию сигналов, модуляцию, кодирование и декодирование, анализ данных, анализ кода.
Комплекс «Акустический полигон» — А-Полигон	Выполнен на основе модификации ТЮК-А и представляет собой полигон для проверки освоения понятий сигнала и модуляции, а также навыков работы с сигналами и обработкой данных на системе, в основе которой лежат те же принципы, но которая представляет собой принципиально другую систему, чем тренировочная. Комплекс — шумоизолированный бокс, внутрь которого возможно помещать до трех излучателей. Кроме того, возможна установка до четырех микрофонов в разные области бокса. Принцип действия аналогичен ТЮК-А, но одновременно передается несколько звуковых файлов с разных излучателей. Система позволяет выполнять задачи с наложением сигналов (подавление помех, деконволюция) и радиолокацией источника.

4.3.5. Описание задачи

Методический комментарий

Заключительный этап проходит на комплексе стендов, взаимосвязанных по компетенциям и моделирующих реальную задачу передачи данных по физическому каналу, анализа данных и создания помехозащищенных протоколов. Итогом работы над задачами является набор протоколов передачи данных с учетом физических особенностей среды.

Участникам ставятся задачи в виде:

- описаний моделей;
- интерфейсов взаимодействия с ними;
- критерия оценки успешности решения отдельных задач.

Командам нужно предоставить решение серии задач в указанном формате, который используется для проведения измерения, формирующего результат решения. Ограничения на решения обусловлены моделью, техническими параметрами стендов.

Итог решения задач — успешно декодированное сообщение; набор программ, реализующих протокол или слежение и выполняющих поставленную задачу с заданной точностью.

На заключительном этапе участники должны решить семь задач. В процессе работы они взаимодействуют со следующим оборудованием:

- задачи №№ 1, 2 — стенд УНКС;
- задачи №№ 3, 4 — О-БТС;
- задачи № 5 — связка стендов О-БТС и ТЮК-А;
- задачи №№ 6, 7 — А-Полигоном.

В задаче № 1 рассматриваются вопросы слежения за подвижным источником сигнала с учетом его диаграммы направленности и стабильной передачи данных в канале.

Цель задачи — отработка навыка взаимодействия с техническими устройствами, куда данные приходят с помехами и наводками, а отладка и исправление сбоев в программе усложнены ввиду удаленности прибора. Потери данных происходят как из-за физического заужения апертуры приемника, так и из-за инверсии поступающих из программы слежения команд.

Решение проверяется на выделенном судейском стенде. Конфигурация параметров инверсии на тренировочных и судейском стендах отличается — на первых искажения пропорционально меньше и реже, на втором — одинаковы для всех команд и не меняются на протяжении всего заключительного этапа.

В задаче № 2 участники работают с телеметрией — технологией, необходимой для взаимодействия с такими системами, где нет физического доступа к оборудованию (например, космические аппараты).

Цель задачи заключается в аппроксимации параметров траектории наблюдаемого «Спутника» по телеметрии, состоящей из набора двоичных статусных регистров неподвижного «Радара», которые снимают во время слежения. Телеметрию необходимо преобразовать как в плане перевода данных в обрабатываемый вид (график отклонения от оси визирования), так и содержательно (перевод отклонения в физическое положение «Спутника»).

Для восстановления модели и проверки гипотез участники работают на тренировочных стендах. Решения отправляются через бот. Измерения траекторий осуществляются на закрепленном судейском стенде. Конфигурация траекторий индивидуальна для каждой команды и неизменна на протяжении всего этапа.

Задача № 3: используя различные методы анализа данных, нужно декодировать сообщения на закрытых от участников дисках. Декодирование происходит по полученным данным с измерений на судейском стенде О-БТС.

С помощью тренировочного стенда участники:

- анализируют выданный им диск с таким же кодом, как на закрытых;
- выявляют паттерны;
- учатся находить их на графиках по данным закрытых дисков.

Решения сдаются через бот. Параметры измерений на судейском стенде закреплены и одинаковы для всех закрытых дисков, но неизвестны участникам.

Задача № 4 посвящена деконволюции — разделению одного сигнала от другого. Цель — разгадать все закрытые сообщения на дисках и излучателях.

На судейском стенде удаленно от участников измеряется одновременно диск с одним типом кода (рисунком на диске), а при измерении излучатель передает сообщение собственным световым кодом (подробнее о кодах в документе «Регламент и условия задач.pdf» (<https://disk.yandex.ru/i/tMQq2zpRMoMaFg>, раздел «Описание кодов»). Таким образом, за одно измерение передается два сообщения.

Для получения измерений участники подают заявку оператору судейского стенда, в которой перечисляются:

- желаемый номер диска;
- номер излучателя;
- цвет излучателя (не влияет на ответ, зависит от того, с каким цветом излучателя участники предпочитают работать на тренировочных стендах).

У каждой команды есть персональный открытый диск для изучения. Решения отправляются через бот.

При решении **задачи № 5** финалисты взаимодействуют сразу с двумя каналами связи (оптическим и акустическим), выстроенными в одну систему.

Работа выполняется в следующем порядке:

1. команде выдается исходное сообщение в битах;
2. участники создают собственные диски, кодируя на них исходное сообщение (способ кодирования — по выбору);
3. диск измеряется на стенде О-БТС;
4. полученный сигнал проходит через заданный медиаконвертер (`converter.py` <https://disk.yandex.ru/d/CNH1-xabozou0Q/converter.py>);
5. выходящий из медиаконвертера файл запускается на стенде ТЮК-А;
6. выходные данные подаются на вход программе-декодировщику, который участники писали самостоятельно и сдавали оператору в начале сеанса;
7. полученное после декодировщика сообщение (биты) сравнивается с исходным сообщением.

Результатом сдачи задания является количество идентичных бит. Для уравнивания условий все измерения проходят на судейском стенде. Цвет излучателя, начальное положение диска О-БТС и параметры усилителей сигнала ТЮК-А участники задают сами.

Задача № 6 посвящена проблеме определения местоположения устройства методом локации. Здесь участники определяют положение излучателя внутри закрытого А-Полигона по измерениям с двух микрофонов, которые команды устанавливали самостоятельно в желаемые позиции.

Сигнал, посылаемый с излучателя, команды также выбирают или пишут само-

стоятельно и передают оператору А-Полигона при измерении.

В задаче № 7 участники работают над реализацией активного шумоподавления — технологией, распространенной в транспорте и аудиооборудовании.

Команда получает результат приема сигнала-помехи, излученного внутри А-Полигона, и на его основании подбирает звуковой сигнал (посылку для ТЮК-А) для подавления помехи при одновременном излучении. Подбор производится посредством четырех серий до двадцати пробных измерений с дальнейшей передачей посылки для оценочного измерения.

Критерием решения является среднеквадратическое отклонение сигнала от тишины (сигнала без излучения помехи). Принцип формирования сигнала-помехи одинаков для всех команд, но варьируются его свободные параметры как между командами, так и между сеансами решения (после оценочного измерения параметры меняются).

Для всех команд условия заданий формулируются одинаково. Они составлены таким образом, чтобы, с одной стороны, быть независимыми, а с другой — образовывать логическую цепочку, однако сдавать их можно в любой последовательности. Количество одновременно решаемых задач определяет сложность командной работы.

Задача 4.3.5.1. УНКС: нестабильное управление (12 баллов)

Условие

Разработка больших технических устройств всегда сопряжена с набором проблем, часть которых можно решить с помощью присутствующего рядом инженера. В случае со спутниками и беспилотными системами это невозможно, что обостряет любые возникающие неполадки: их сложнее отладить и исправить.

В этой задаче организован инфракрасный канал связи на стенде УНКС. Среда передачи благоприятная — помех и шумов нет, данные передаются и принимаются без ошибок. Для качественной работы канала связи достаточно лишь организовать точное слежение «Радара» за «Спутником».

При этом существует проблема: протокол управления «Радаром» реализован без учета помех, из-за чего «Радар» управляется нестабильно. А именно, периодически значения смещения оси визирования «Радара» приходят с погрешностью. В некоторые моменты (не зависящие от ранее упомянутых) отправленные команды инвертируются: вместо поворота в одну сторону «Радар» начинает вращаться в обратную, а вместо остановки — вращаться в случайном направлении. Кроме того, скорость в это время тоже приходит с погрешностью.

Задача: реализовать программу слежения (**tracker**) для «Радара», выполняющую максимально точное слежение за «Спутником» в условиях нестабильного управления. Среда передачи не имеет шумов, передача данных осуществляется без дополнительного кодирования, то есть кодер и декодер встроены и передают данные без изменений.

Программа слежения пишется на C, C++, Java или Python 3 в одном файле **tracker**.

Примеры трекеров размещены в директории `UserExamples` в файлах с названием `tracker`, например, трекер на языке Python находится в файле `tracker.py` (<https://disk.yandex.ru/d/CNH1-xabozou0Q/UserExamples/>).

Ниже представлен пример трекера на языке Python 3.

Python

```

1  from client2server import client2server
2
3  class tracker:
4      def run(tracklog):
5          c2s = client2server()
6          i = 0
7          while i == 0:
8              status = c2s.getStatus()
9              dx = int(status) & 0x0fff
10             if dx > 2848:
11                 dx = dx - 4096
12                 # tracklog.write(f"{dx}".encode())
13             if abs(dx) < 500:
14                 if abs(dx) < 10:
15                     c2s.moveStop()
16                 else:
17                     if dx > 0:
18                         c2s.moveLeft(4)
19                     else:
20                         c2s.moveRight(4)

```

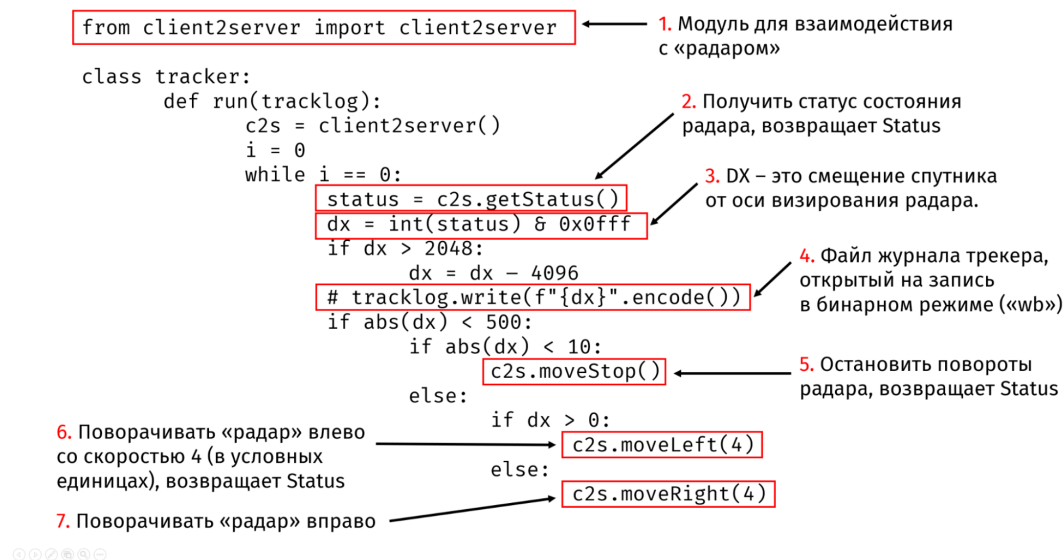


Рис. 4.3.6. Пример трекера с пояснениями

Описание интерфейса управления слежением приведено в файле `0_readme.txt` в этой же директории.

Порядок работы:

1. Команда обрабатывает решение на тренировочных стендах со схожей конфигурацией задачи (частота шума, и траектория отличаются), после чего заявляет зачетную попытку в рамках очередного слота.

Пример заявки:

```
1 #заявка з1 к2
```

2. Представитель команды приносит оператору программу-трекер на флешке. Файл должен иметь имя `tracker.[расширение]`, например, `tracker.py`. Оператор загружает программу на стенд и запускает передачу данных со слежением.
3. По завершении передачи оператор фиксирует процент слежения в рейтинговой таблице. По запросу команды оператор может скопировать файл журнала на их флешку.

При возникновении ошибки компиляции программы оператор откладывает заявку для исправления проблемы. Отложенная заявка переносится в конец очереди. Если команда не приносит решение после переноса, ее заявка в текущем слоте аннулируется.

Критерии оценивания

Максимальный балл за задачу — 12.

В зачет идет лучшая попытка. Оценивается точность передачи (программным путем). Нижний порог — 25% точности передачи.

Балл за задачу начисляется по формуле

$$y = N \cdot \frac{x - k}{100 - k},$$

где N — балл за задачу, k — значение точности передачи при запуске на демонстрационном трекере (нижний порог), x — процент, показывающий точность переданного файла при запуске на трекере, представленном командой.

Иначе говоря, N баллов — максимальное количество баллов при 100% точности передачи контрольного файла по каналу связи, а 0 баллов — точность передачи контрольного файла по каналу связи с использованием стандартного трекера.

Решение

На тренировочном стенде команда может ознакомиться с упрощенной версией задачи и провести пробные сеансы слежения для установки характера искажений. Для этого необходимо извлечь данные статусного числа в журнал слежения, а затем проанализировать, в первую очередь получив время, отклонение от оси визирования (DX).

Работа с искажением DX строится на реализации слежения, стабильного к резким изменениям, например, за счет фильтрации значений DX.

Инверсия направления вращения фиксируется за счет сохранения отправленного направления и сравнения его со значением регистра статусного числа, отвечающего за направление вращения, и компенсируется ответной инверсией.

Искажение скорости вращения фиксируется через резкое направленное изменение производной DX, компенсируется путем изменения скорости вращения. В общем случае скорость уменьшается для стабилизации вращения.

Дальнейшая работа выполняется алгоритмом слежения, который может строиться на пропорциональном или ПИД-регуляторе, параметры которого подбираются итерационно за счет организованной передачи телеметрии с судейского стенда.

Задача 4.3.5.2. УНКС: телеметрия (8 баллов)

Условие

Телеметрия — это технология автоматического наблюдения, сбора и передачи данных, необходимых для организации работы технической системы. В частности, для сбора данных с действующих космических аппаратов и спутников, где нет физического доступа к оборудованию.

Важная часть работы со спутником — это знание траектории его перемещения. Иногда она заведомо известна, но в данном случае возможности ограничены. В распоряжении участников телеметрия, снимаемая с зафиксированного «Радара» в виде набора статусных чисел — 128-битных чисел, в которых закодировано текущее состояние «Радара» (описание формата статусного числа приведено в файле `UserExamples/0_readme.txt` <https://disk.yandex.ru/d/CNH1-xabozou0Q/UserExamples/>).

«Радар» не двигается во время снятия траектории, но известно, что при перемещении «Спутник» не покидает область видимости камеры «Радара».

Задача: максимально точно восстановить три траектории «Спутника» по данным телеметрии.

В ответе необходимо указать четыре вещественных числа через пробел:

- большая полуось орбиты планеты, см;
- период вращения планеты, с;
- радиус орбиты спутника, см;
- период вращения спутника, с.

Например, 20,4 15,3 2,2 10,1.

Заведомо известно, что фокусное положение звезды всегда в центре оси.

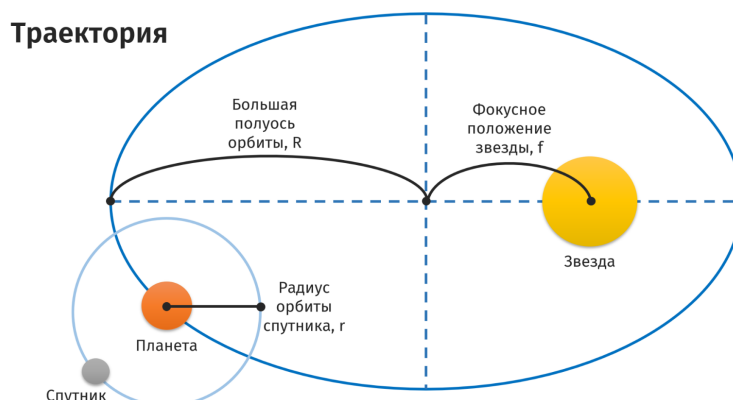


Рис. 4.3.7

Таблица 4.3.2

Название параметра	Описание
Planet_Semimajor_axis(cm)	Большая полуось орбиты планеты R , см
Planet_orbital_period(sec)	Период обращения планеты вокруг звезды P , с
Planet_Focus(cm)	Фокусное положение звезды относительно центра оси f , см
Sputnik_Radius_orbit(cm)	Радиус круговой орбиты спутника вокруг планеты r , см
Sputnik_orbital_period(sec)	Период обращения спутника вокруг планеты p , см

Параметры траектории

На схеме выше «Радар» располагается на значительном удалении от звезды и работает с проекцией траектории на горизонтальную ось.

Порядок работы:

1. Участники заявляют зачетную попытку в рамках очередного слота и указывают номер траектории для измерения.

Пример заявки:

```
1 #заявка з2 к3 т1
```

где $t[\text{номер}]$ — выбранная траектория для измерения.

2. Оператор запускает слежение и снятие телеметрии. Во время слежения команда может присутствовать рядом со стендом, но использование фото- и видеосъемки категорически запрещено.
3. По завершении слежения оператор передает файл телеметрии на флешку, предоставленную командой.
4. Команда анализирует телеметрию и отправляет ответ через бот-кодоприемник. Пример сообщения в бот: о3-1 20.4 15.3 2.2 10.1 (шаблон о[номер команды]–[номер траектории] [параметры траектории]).

Критерии оценивания

Максимальный балл за задачу — 8.

Сравнение траектории с эталонном осуществляется по четырем параметрам. При отклонении каждого параметра не более, чем на 1 см или 1 с, траектория засчитывается на полный балл. При отклонении параметров решение отклоняется. У участников по три попытки на каждую траекторию.

Гарантируется, что радиус орбиты планеты всегда больше радиуса орбиты спутника. Если траектория состоит из одной орбиты, укажите 0 во втором радиусе, тогда второй период не будет учитываться при проверке.

Вес первой траектории — 20%, второй и третьей — по 40%.

Решение

Файл телеметрии содержит набор 128-битных статусных чисел в двоичном виде, то есть файл делится на пакеты по 16 байт. Согласно описания формата числа, его биты 0–11 содержат знаковое 12-битное целое число — смещение спутника на камере «Радара», которое измеряется в пикселях. Кроме того, 20–36 биты содержат беззнаковое целое время с начала работы «Радара», по которому можно отследить изменение статусного числа и соответственно DX.

При извлечении и сопоставлении двух временных рядов образуется график смещения «Спутника» относительно центральной оси стенда. При этом важно отсечь часть, соответствующую калибровке «Спутника» (в среднем она продолжается до 24 с или около 300 строк). По нему путем аппроксимации (например, оптимизационными методами) восстанавливается период в миллисекундах и амплитуда в пикселях.

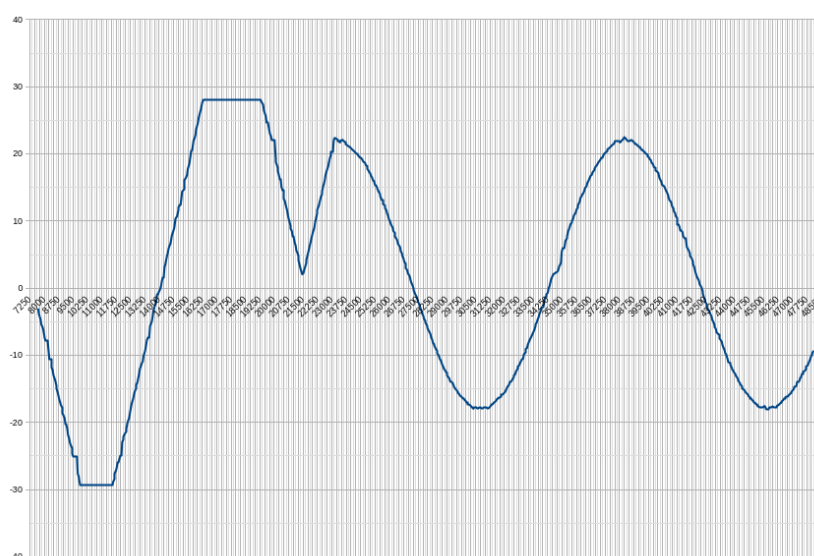


Рис. 4.3.8. Пример траектории 1

Для перевода пикселей в сантиметры необходимо восстановить геометрическую модель стенда. Для этого можно использовать тренировочный стенд, слежение за «Спутником» без вращения и с записью телеметрии, а также измерительные приборы (линейка, рулетка).

Нескольких точек достаточно для определения линейной зависимости геометрического положения «Спутника» от DX: $x = -0,1309 \cdot DX + 1,6778$.

Задача 4.3.5.3. О-БТС: восстановление пароля (13 баллов)

Условие

В практике любой компании, существующей длительное время, есть процесс устаревания информации, в том числе программного кода. И если не уделять этому должное внимание, появляется «легаси» — артефакты от прошлых разработчиков, представление о которых имеют только они сами. Часто «легаси» работает, но никто не знает, почему, и это весьма опасно.

Смоделируем ситуацию: существует информационная система, разработанная давно и в целом не требующая к себе внимания. Нужно перенастроить ее под новые стандарты, для чего требуются пароли доступа. Как известно, их нужно хранить в надежном месте.

Прошлые сотрудники записали эти пароли на диски с кодом `nto10.t[номер]` («Регламент и условия задач.pdf» (<https://disk.yandex.ru/i/tMQq2zpRMoMaFg>), раздел «Описание кодов») для стенда О-БТС. Но найти удалось только один диск (его экземпляр в распоряжении у каждой команды), и пароль с него не подошел.

К счастью, один из сотрудников сделал резервную копию, измерив все диски на стенде. Сделал он это за один подход, то есть на одном стенде с одинаковыми параметрами. Параметры измерения, включая цвет излучателя, утеряны. Но известно, что на всех дисках используются те же буквы, которые есть на открытом диске.

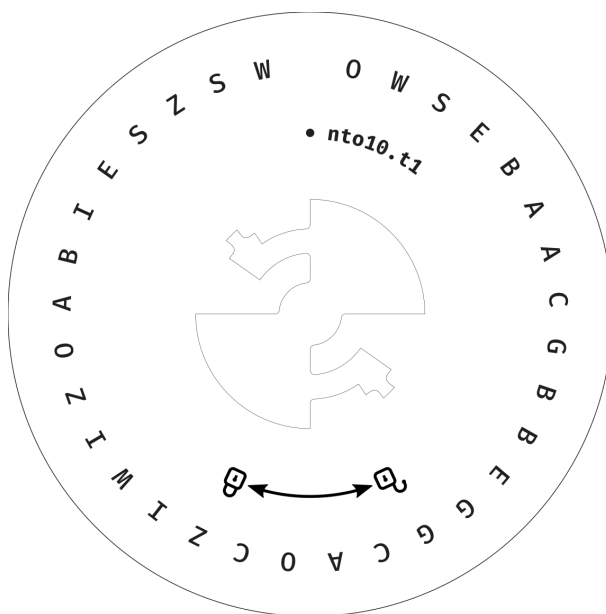


Рис. 4.3.9. Пример диска `nto10.t`

Задача: с помощью этих измерений восстановить 25 паролей, закодированных на всех закрытых дисках. В распоряжении команды есть открытый диск с примером пароля.

Порядок работы

Участники исследуют выданный команде диск и декодируют данные с закрытых дисков. Закрытые диски лежат на рабочем столе терминала команды в папке «Задача 3 — Закрытые диски». Результаты декодирования передаются в бот-кодоприемник.

Пример сообщения в бот:

`t3-12 ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEF`
шаблон `t[номер команды]-[номер диска][сообщение с диска]`.

Критерии оценивания

Максимальный балл за задачу — 13.

Балл за каждое сообщение с закрытых дисков выставляется по результату автоматической проверки сообщения через бот-кодоприемник. Сравнение с правильным ответом осуществляется по расстоянию Левенштейна. Полное совпадение соответствует полному баллу за сообщение (0,52 балла за один полностью решенный диск), каждая единица ошибки отнимает 20% от полного балла за один диск. На каждое сообщение дается по три попытки.

Решение

Задача подразумевает выявление и сопоставление паттернов, соответствующим каждой букве. Для их восстановления предназначен открытый диск. Его необходимо измерить на тренировочном стенде, при этом нужно подобрать параметры измерения для получения выраженных паттернов. При подборе необязательно стремиться к тому, чтобы получить параметры, максимально близкие к судейскому стенду, важно именно получение различных паттернов, соответствующих буквам.



Рис. 4.3.10. Алфавит кода дисков nto10.t



Рис. 4.3.11. Диск на 32 буквы

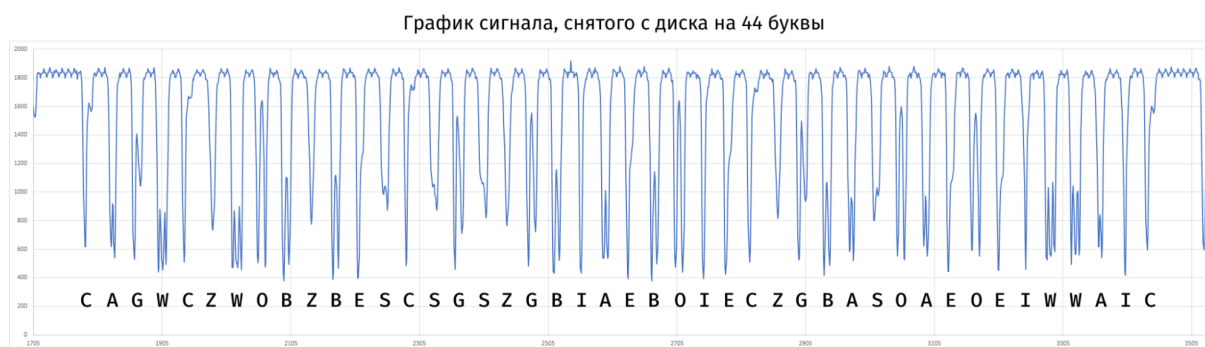


Рис. 4.3.12. Диск на 44 буквы

После получения паттернов производится их сопоставление с закрытыми данными. Это можно производить вручную или программно: разделение сигнала на отдельные отрезки тривиально после фильтрации статического шума, сопоставление можно выполнять различными методами (в том числе через отклонения).

Задача 4.3.5.4. О-БТС: разделение каналов (14 баллов)

Условие

В телекоммуникации существует задача разделения каналов, то есть передачи сразу нескольких сигналов по одной линии связи. Инженеры представили много различных вариантов ее решения.

Производится двухпоточковая передача данных по оптическому каналу стенда О-БТС.



Рис. 4.3.13. Пример диска nto10.d

Первое сообщение формируется модулирующим диском с кодом nto10.d[номер] («Регламент и условия задач.pdf» (<https://disk.y>

[andex.ru/i/tMQq2zpRMoMaFg](https://disk.yandex.ru/i/tMQq2zpRMoMaFg)), раздел «Описание кодов»), который вращается с фиксированной минимальной скоростью («Быстрый старт.pdf» <https://disk.yandex.ru/i/1KcWVMK5RoBLcQ>).

Второе сообщение формируется изменением яркости излучателя в течение времени согласно коду `nto10.e`.

Периоды вращения диска и период передачи сообщения излучателем не совпадают. Измерение канала начинается в произвольный момент, снимается 4 500 измерений с интервалом в 2 мс.

Задача: восстановить сообщения с дисков и излучателя, используя результаты удаленных двухпоточковых передач. В каждый слот можно запросить до трех передач, в каждой из которых выбрать один из десяти дисков: `d1–d10`, одно из двадцати сообщений излучателя: `e1–e20`, и модуль излучателя (красный, желтый, зеленый или синий). Диски и сообщения излучателя закрыты и перемешаны, т. е. у каждой команды под одним номером будет свое сообщение.

Порядок работы

Взаимодействие с судейским стендом осуществляется исключительно оператором, без присутствия участников.

Порядок действий:

1. Вместе с заявкой на задачу участники передают до трех комбинаций из номера сообщения на излучателе, номера диска и цвета излучателя.

Комбинация имеет вид: `излучатель [номер], диск [номер], [цвет излучателя]`.

Пример заявки с тремя комбинациями:

```
1 #заявка 34 к31
```

- 1) излучатель 5, диск 1, желтый;
 - 2) излучатель 6, диск 10, зеленый;
 - 3) излучатель 7, диск 3, красный.
2. Оператор выполняет измерения с фиксированными параметрами: диафрагма 0,5 мм, интервал измерения 2 мс, число измерений 4500, скорость вращения 1, базовая амплитуда блока излучателя сигнала зависит от выбранного цвета (красный — 400, зеленый — 48, синий — 35, желтый — 400). Результаты измерений передаются команде по готовности — оператор вызывает команду в чате инженерной задачи, участник от команды должен подойти к оператору с флешкой.
 3. Команда производит анализ результатов, определяет сообщения и отправляет их в бот-кодоприемник.

Пример сообщения в бот: `e3-1 ABCABCSABCSABCSABCSABCSABCSABCS`
шаблоны сообщений:

`e[номер команды]-[номер сообщения][сообщение с излучателя];`

`d[номер команды]-[номер сообщения][сообщение с диска].`

Критерии оценивания

Максимальный балл за задачу — 14.

Балл за каждое сообщение выставляется по результату автоматической проверки сообщения через бот-кодоприемник. Одна точно декодированная последовательность символов сообщения диска или излучателя дает 0,4667 балла, всего последовательностей 30. Сравнение с правильным ответом осуществляется по расстоянию Левенштейна. Полное совпадение соответствует полному баллу, каждая ошибка отнимает 20% этого балла. На каждое сообщение дается по три попытки.

Решение

Первый этап — измерение открытого диска на разных яркостях излучателя и выявление закономерностей изменения яркости в зависимости от уровня заливки диска. Далее производится анализ полученных измерений.

При наложении сообщения излучателя на диск смешиваются два сигнала: четырехуровневый (если считать стартовое условие) и двухуровневый. Умножение двух сигналов дает набор из восьми уровней амплитуды.



Рис. 4.3.14. Последовательное изменение яркости излучателя

Разделение этих уровней на данных измерения выполняется вручную:

1. построение восьми горизонталей;
2. визуальное выявление переходов;
3. разделение двух периодических сигналов;
4. снятие сообщения;

или программно:

1. поиск пиков производной;
2. выявление двух периодических сигналов;
3. разделение их уровней;
4. снятие сообщения.



Рис. 4.3.15. Измерение задача 4

Задача 4.3.5.5. О-БТС + А-ТЮК: биатлон (17 баллов)

Условие

Передача данных может выполняться в разных средах, но периодически возникает необходимость организовать канал, переходящий из одной среды в другую, например, из оптоволокну в витую пару. Для этого используются медиаконвертеры.

Дан оптоакустический канал связи, состоящий из О-БТС и ТЮК-А, объединенных программой-медиаконвертером. Иначе говоря, сигнал, заданный диском, передается через акустический канал. Судейские стенды такие же, как и тренировочные. Расстояние от микрофона до излучателей на ТЮК-А составляет 70 см, расстояние между микрофонами — 15 см.

Задача: передать по этому каналу до 40 бит информации так, чтобы полученное сообщение (биты) максимально соответствовало исходному.

Исходное сообщение наносится на диск и измеряется на стенде О-БТС. Результат измерения обрабатывается программой-медиаконвертером, и полученный файл передается по акустическому каналу на А-ТЮК. Принятый сигнал отправляется в программу-декодировщик (декодер), который восстанавливает из сигнала биты. Полученное после декодера сообщение сравнивается с исходным, и результат решения оценивается исходя из количества правильно переданных бит.

Порядок передачи сообщения представлен на схеме работы (см. рис. 4.3.16).

Для сдачи задания нужно предоставить на флешке оператору свой декодер, после чего оператор выдает исходное сообщение — 40 бит, которое необходимо передать оператору на диске. Изображение выполняется доступными средствами: в распоряжении каждой команды есть диски без изображения (3 шт.), изолянт и ножницы, у организаторов можно запросить шесть листов прозрачной пленки для печати на принтере.

Оператор измерит переданный диск на судейском стенде О-БТС с фиксированными параметрами:

- диафрагма 0,5 мм;
- интервал измерения 2 мс;
- число измерений 2000;
- скорость вращения 1;

- интервал измерения 2 мс;
- базовая амплитуда блока излучателя сигнала зависит от выбранного цвета (красный — 400, зеленый — 48, синий — 35, желтый — 400).

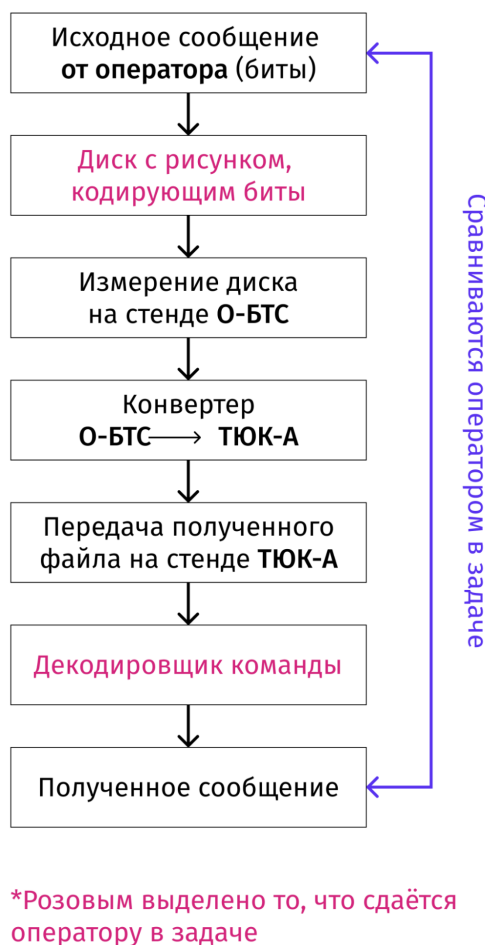


Рис. 4.3.16. Схема работы в задаче 5

Полученный сигнал с помощью функции-медиаконвертера (`converter.py` <https://disk.yandex.ru/d/CNH1-xabozou0Q/converter.py>) нормализуется и удлинняется в три раза путем тройного повторения каждого измерения, после чего передается в качестве посылки через ТЮК-А. Результат передачи вводится в ранее сданный декодер.

Восстановленное декодером сообщение сравнивается с исходным, и результат попытки определяется числом корректно переданных бит.

Декодер пишется на С, С++, Java или Python 3 в одном файле `sound_decode`. Входные данные программа читает из входного потока (`stdin`), а выходное сообщение — в выходной поток (`stdout`). Чтение и запись файлов недопустимы. Входные данные представлены в формате CSV-файла ТЮК-А, при этом во входных данных столбец `Input` будет засекречен: все значения в этом столбце будут равны `—1`. Выходное сообщение выводится как текстовая последовательность нулей и единиц (например, `0101010110`).

Порядок работы:

1. Участники дают заявку на задачу в очередной слот. В порядке очереди ко-

манда приносит оператору программу-декодер и в ответ получает файл с сообщением для передачи.

Пример заявки для сдачи декодера:

```
1 #заявка з5 к3
```

При возникновении ошибки компиляции оператор откладывает заявку для исправления проблемы. Отложенная заявка переносится в конец очереди. Если команда не приносит решение после переноса, ее заявка в текущем слоте аннулируется.

2. Команда самостоятельно создает на диске изображение, соответствующее сообщению.
3. Сдача диска оператору производится в порядке живой очереди, с выбором цвета блока излучателя сигнала и выставлением параметров усилителя ТЮК-А. Команда выставляет диск на судейском стенде, после оператор запускает измерение с фиксированными параметрами из условия задачи. Полученный сигнал автоматически преобразуется с помощью медиаконвертера и передается через ТЮК-А. На полученном сигнале запускается декодер команды. Выходной файл декодера сравнивается с исходным.
4. Команда может неограниченно сдавать диски на одном декодере, смена декодера осуществляется в рамках новой попытки и сопровождается генерацией нового сообщения.

Критерии оценивания

Максимальный балл за задачу — 17.

Балл за решение выставляется в результате автоматического сравнения с исходных сообщением по расстоянию Левенштейна. За успешную полную передачу сообщения выставляется полный балл. Каждая ошибка снимает 5% полного балла

Решение

Задача проверяет способность переносить сигнал из одной среды в другую. Команда имеет на руках программу-медиаконвертер, за счет чего способна проводить эксперименты с собственным чистым диском, а именно, подбирать паттерн изображения, способный формировать различимый звук, который можно зафиксировать программно.

Оптическая среда О-БТС регистрирует значение уровня, поэтому необходимо создать монохромное изображения. Акустическая среда А-ТЮК регистрирует изменение значения уровня, порождающее звуковую волну. Соответственно, наиболее эффективным будет кодирование сообщения за счет изменений уровня яркости.

Например, создание набора импульсов (щелчков) или воспроизведение широтно-импульсной модуляции. При правильном подборе частоты полос на печатном изображении диска (см. пример ниже) возможно добиться громкого звучания сигнала на А-ТЮК, регистрируемого при амплитудной или частотной демодуляции. Для большей точности декодирования можно использовать разные параметры усилителя.

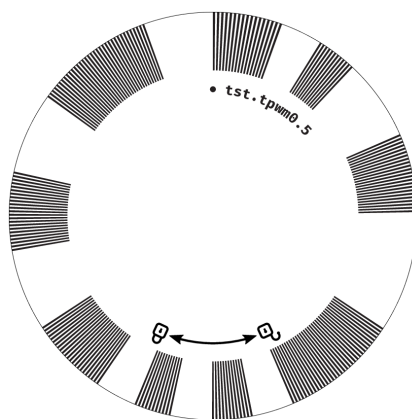


Рис. 4.3.17. Диск-пример решения задачи 5

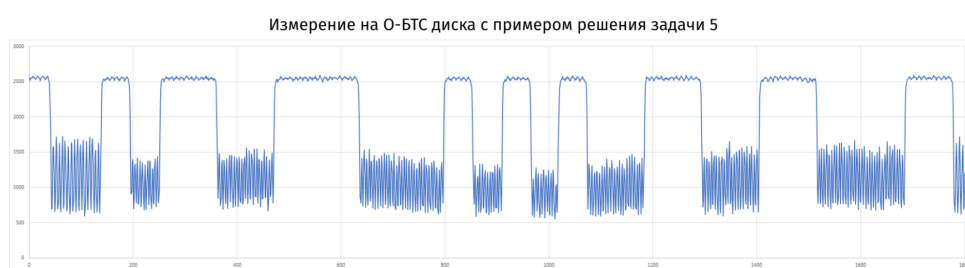
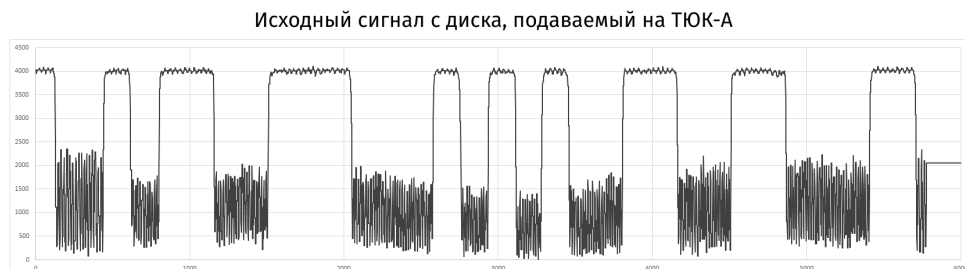
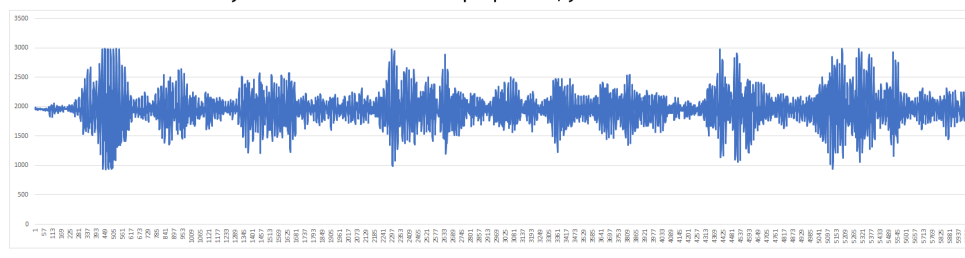


Рис. 4.3.18. Измерение на О-БТС (задача 5)



Полученный сигнал на микрофоне А, усиление сигнала x200



Полученный сигнал на микрофоне Б, усиление сигнала x300

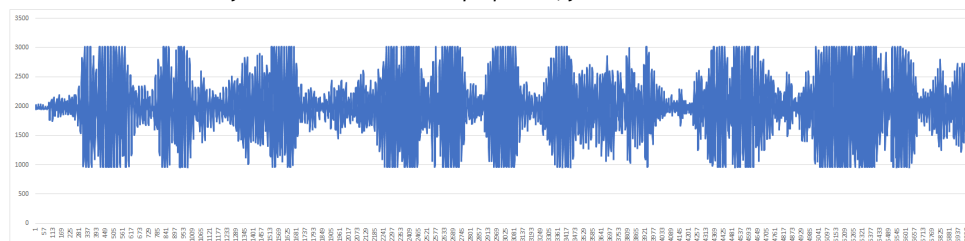


Рис. 4.3.19. Прогон данных с диска на ТЮК-А

Задача 4.3.5.6. А-Полигон: аудиолокация (15 баллов)

Условие

Нередко для эффективной передачи данных требуется знать положение источника сигнала, в некоторых случаях — активно на него направляться. В этой задаче участникам предстоит определить положение источника звука внутри акустического полигона.

В случайную ячейку поля выставляется излучатель на стойке. Высота излучателя на уровне микрофонных разъемов; погрешность ± 1 см. Излучатель всегда ориентирован в сторону микрофонов «С-1»–«С-4» («Быстрый старт.pdf» <https://disk.yandex.ru/i/1KcWVMK5RoBLcQ>).

Задача: восстановить положение излучателя с точностью до ячейки поля.

Посылка длиной 125 мс предоставляется командой оператору. Перед каждой передачей команда выставляет два микрофона в любые два разъема полигона на свое усмотрение. Для передачи сигнала используется такой же ТЮК-А, как и тренировочный. Конфигурация усилителей зафиксирована организаторами и одинакова для обоих микрофонов — установлен джампер x2. Восстановление положения излучателя производится командой на основе измерения двух закрытых передач на комплекте ТЮК-А.

В рамках задачи доступны тренировочные и зачетные попытки.

Для каждой зачетной заявки выставляется новое положение излучателя, и на его определение дается две попытки.

Для тренировочных заявок положение излучателя считается отдельно от зачетных и не меняется, пока команда сама об этом не попросит. Количество попыток на разгадывание тренировочной заявки не ограничено.

В обоих случаях положение излучателя не раскрывается. За один слот можно сделать либо тренировочную заявку, либо зачетную.

Обратите внимание: при передаче сигнала присутствует запаздывание (его значение постоянно, но одинаково между комплектами), а значение скорости звука сложнее, чем обычно дается в справочниках.

Порядок работы

Команды работают на тренировочных комплектах ТЮК-А и заявляют попытки на полигон в рамках слота. Взаимодействие с полигоном осуществляется следующим образом: участники в присутствии оператора устанавливают микрофоны в желаемые позиции. После этого оператор проводит два измерения, результат которых в виде CSV-файлов передается участникам на флешку.

Порядок действий:

1. Команда делает заявку на попытку и указывают ее тип: тренировочная или зачетная. Вместе с заявкой участники передают посылку и ставят в желаемые позиции два микрофона. Файл посылки передается оператору на флешке. Файл должен иметь имя `input.txt` и быть длиной 6 000 тактов.
Пример тренировочной заявки:

```
1 #заявка з6 к3 Т
```

Пример зачетной заявки:

```
1 #заявка з6 к3 З
```

2. После заявки на задачу оператор инициализирует сеанс работы, закрытую для всех участников позицию ставит излучатель в случайное положение (оно записывается на судейском терминале) и производит три передачи посылки. Принятые сигналы передаются команде для изучения.
3. Команда анализирует принятые сигналы, формирует ответ и сдает его через оператора в порядке живой очереди. В случае верного решения выставляется балл. В случае ошибки команда может сдать ответ еще раз, после чего попытка закрывается.
4. После закрытия попытки необходимо сделать повторную заявку.

Критерии оценивания

Максимальный балл за задачу — 15.

Считается по лучшей попытке. За точное определение позиции излучателя на сетке выставляется полный балл, если ответ дан с первой попытки; 50% — если со второй. После второго неверного ответа выставляется 0%.

Решение

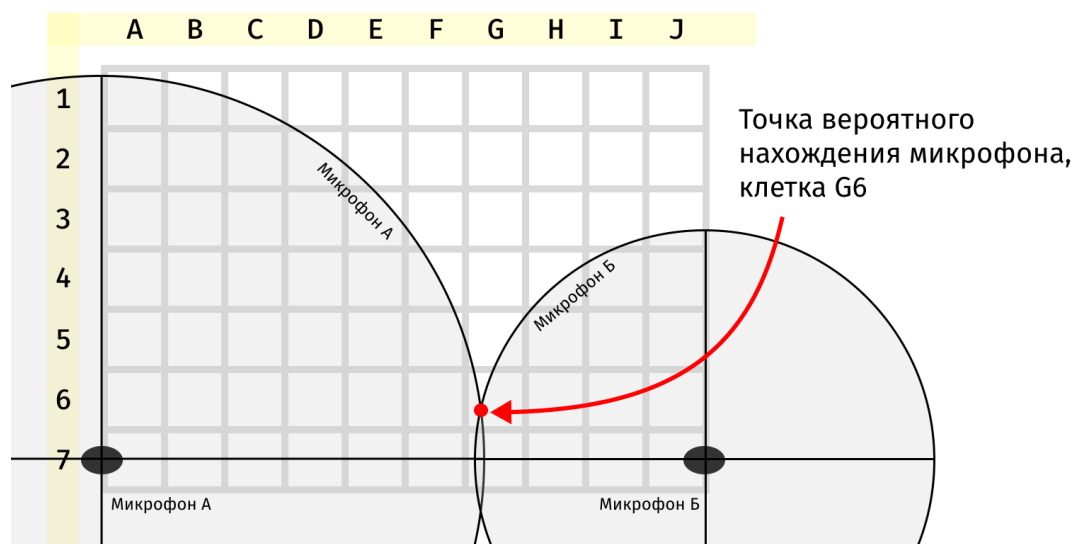
Наилучшее положение для измерения — максимально широкая база из микрофонов, стоящих в противоположных стенках А-Полигона друг напротив друга. Положение имеет смысл изменить на противоположное для учета возможной дисторсии при излучении рядом с микрофонами, а также для получения дополнительных точек.

Восстановление позиции излучателя требует максимально точного восстановления модели звука. Фактически решением является выполнение триангуляции, для которой производится преобразование задержки между отправкой и приемом посылки на расстояние от излучателя до микрофона.

Задержка измеряется по смещению характерных пиков сигнала-посылки и сигнала на микрофоне. Смещение измеряется в тиках и переводится в миллисекунды с помощью первого столбца CSV-файла, либо через частоту дискретизации 48 кГц.

Далее необходимо восстановить систематическую задержку и скорость звука. Для этого на тренировочных стендах следует провести серию измерений на разных расстояниях и их дальнейший анализ.

После преобразования расстояний строятся две (или четыре) окружности, пересечение которых указывает на вероятную точку нахождения микрофона.



Диаметр для микрофона =
 Время до появления звука на микрофоне * Скорость звука

Рис. 4.3.20. Пример решения задачи 6

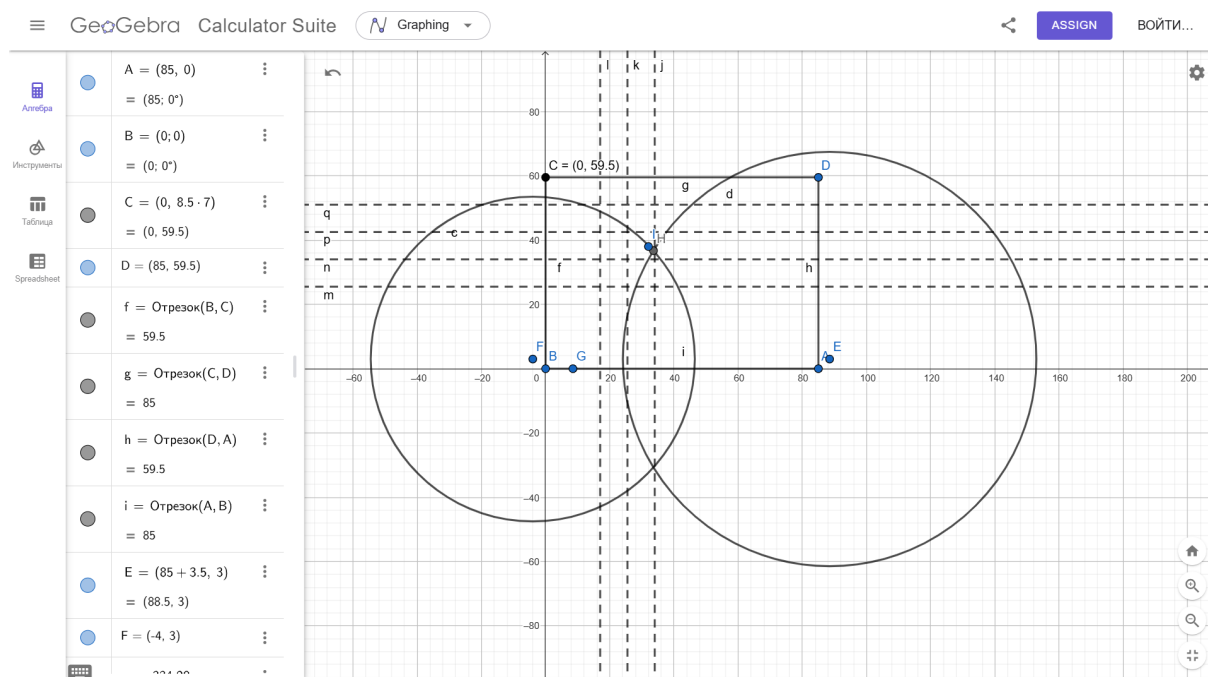


Рис. 4.3.21. Пример более подробного решения задачи 6

Задача 4.3.5.7. А-Полигон: шумоподавление (21 балл)

Условие

Передача данных неразрывно связана с обеспечением помехоустойчивости канала связи. К борьбе с шумами можно подходить разными способами, и один из

них — активное шумоподавление. Его работу можно ощутить собственными ушами: на рынке существует много наушников с ANC.

Дан канал связи в среде с помехой в виде стороннего сигнала. Этот сигнал детерминирован и обладает определенным законом изменения со свободными параметрами, значение которых неизвестно.

Задавить помеху передатчиком невозможно — его мощность ограничена. Протокол передачи задан за пределами канала связи, поменять его тоже нельзя. Следовательно, для защиты канала связи от помехи нужно прибегнуть к шумоподавлению. Здесь оно представлено в виде посылки, которая отправляется с излучателя одновременно с полезным сигналом. Для настройки шумоподавления полезный сигнал отключен, и вместо него передается тишина (постоянная амплитуда).

Задача: составить такой сигнал шумоподавления для заданной помехи, чтобы получить на приемнике сигнал, максимально близкий к фоновому звуку (сигналу без работы излучателей). Критерием решения задачи будет среднеквадратичное отклонение принятого сигнала от фонового звука.

Внутри акустического полигона установлены два излучателя. Один будет излучать помеху, второй — сигнал шумоподавления. Позиции обоих излучателей зафиксированы на протяжении всего инженерного тура. В задаче используется только один микрофон, его положение также зафиксировано. По запросу команды оператор может открыть А-Полигон и показать конфигурацию поля.

Параметры помехи отличаются для каждой команды и меняются между сеансами работы. Излучатели и микрофоны используются те же самые, что и в А-ТЮК. Частота дискретизации сигнала отличается, она составляет 24 кГц.

Порядок работы

Взаимодействие с полигоном осуществляется исключительно оператором, без присутствия участников. Но по просьбе команды оператор может открыть крышку полигона и дать участникам рассмотреть установку (без использования фото и видеосъемки).

Порядок действий:

1. Команда делает заявку на задачу в очередной слот.

Пример заявки:

```
1 #заявка з7 к3
```

2. После заявки на задачу оператор инициализирует сеанс работы и производит три передачи сигнала без шумоподавления. Архив с принятыми сигналами передается команде для изучения.
3. Далее команда может выполнить четыре тренировочные серии измерений. Каждая серия представляет собой архив с именем `data_#.zip`, где # — номер серии от 1 до 4. Внутри архива находится до двадцати файлов-посылок в формате А-ТЮК, содержащих сигналы шумоподавления для тестирования. Оператор принимает архив, производит передачу серии сигналов и в следующий слот возвращает архив с результатами передачи и значениями метрики.
4. После тренировочных серий (или раньше) команда передает файл `final.txt` в формате А-ТЮК с сигналом шумоподавления для зачетной серии измерений. Сигнал передается двадцать раз, и для расчета результата берется медианное

значение метрики. Оператор сообщает команде балл за задачу и возвращает архив с результатом зачетной серии.

5. После зачетной попытки необходимо оставить новую заявку на задачу. Конфигурация помехи изменится.

Критерии оценивания

Максимальный балл за задачу — 21.

Балл за задачу выставляется по результату автоматической проверки результатов зачетной серии измерений. В качестве метрики сравнения используется среднее квадратичное отклонение сигнала от тишины. За нижний порог берется 70% от метрики для сигнала без шумоподавления. Если полученный сигнал отличается от идеального (полная тишина без помехи) не более, чем на 15%, ставится полный балл. Балл за значение метрики между нижним и верхним порогом рассчитывается пропорционально.

Решение

В задаче использовалась следующая конфигурация помехи: несколько импульсов синусоидального сигнала с одинаковой частотой и амплитудой, но разной длительностью и расстоянием между импульсами. Фаза синусоидального сигнала общая для всего сигнала помехи. При этом начало и конец каждого импульса совпадали с половиной фазы (для отсутствия резких скачков уровня и вызываемых ими шумов).

На основе передачи сигнала без шумоподавления восстанавливается частота синусоидального сигнала помехи. Восстановление частоты можно осуществлять различными способами, в т. ч. с помощью обратного преобразования Фурье.

При наличии информации о физической конфигурации излучателя и микрофонов (см. ниже), геометрических параметров установки (она соответствует А-Полигону из задачи № 6 с точностью до 1 см) и параметров прохождения сигнала (систематическая задержка и скорость звука вычисляются в задаче № 6) можно получить начальную оценку смещения и фазы сигнала. Кроме того, за счет экспериментов на тренировочных комплектах можно восстановить характеристику затухания сигнала в зависимости от расстояния и получить более точную начальную оценку амплитуды. Смещения импульсов относительно друг друга вычисляются тривиально, достаточно определить смещение первого импульса.

Далее производятся четыре серии по двадцать измерений, в рамках которых аппроксимируются параметры сигнала шумоподавления. Каждое измерение сопровождается значением метрики, потому можно использовать серии для оптимизации значения параметров.

Одним из возможных подходов будет использовать серии измерений для оптимизации:

1. Смещение сигнала (визуальный критерий — полное наложение сигнала шумоподавления и помехи, отсутствие левого «хвоста» из одиночного гармонического сигнала).
2. Фаза (критерий — минимизация или максимизация СКО, второе означает попадание в резонанс и требует взятия противофазы).

3. Амплитуда (критерий — минимизация СКО, а также сглаживание сигнала).

Четвертая серия используется для итогового уточнения параметров с малым шагом.

Фото конфигурации расположения излучателей приведено в разделе «Оборудование и программное обеспечение» и на рис. 4.3.5 «А-Полигон внутри».

4.3.6. Материалы для подготовки

Курсы от разработчиков профиля ТБС

1. Курс «Технологии беспроводной связи» познакомит с базовыми понятиями беспроводной связи и поможет погрузиться в тематику профиля: <https://onti.polyus-nt.ru/course/view.php?id=17>.
2. Курс «Тематические разборы задач профиля ТБС»: <https://onti.polyus-nt.ru/course/view.php?id=3>. В курсе собраны все задачи второго тура профиля за 2016/2017–2020/2021 гг. по темам.
3. Видеоразборы задач сезона 2024–25 приведены в курсе «Разбор задач второго этапа профиля ТБС НТО 24/25»: <https://onti.polyus-nt.ru/course/view.php?id=23>.

Материал про помехи и шумы

4. Статья «Связь и радионавигация. Причины ошибок»: <https://cirspb.ru/blog/ustroystvo-sudov/prichiny-oshibok/>.

Материалы про деконволюцию

5. Статья «Что такое деконволюция?»: <https://temchromatinlab.wordpress.com/2016/10/11/что-такое-деконволюция/>.
6. Статья «Восстановление расфокусированных и смазанных изображений»: <https://habr.com/ru/articles/136853/>.
7. Давыдов А. В. «Цифровая обработка сигналов: Тематические лекции»: <http://www.geokniga.org/books/9445>.

Материал про сочетание оптического и акустического каналов

8. Статья «Оптико-акустический канал»: https://bstudy.net/670914/informatika/optiko_akusticheskiy_kanal.

Материалы про кодирование данных

9. Статья «Помехоустойчивое кодирование с использованием различных кодов»: <https://habr.com/ru/articles/111336/>.
10. Статья «Коды Рида — Соломона. Часть 1 — теория простым языком»: <https://habr.com/ru/companies/yadro/articles/336286/>.
11. Статья «Коды Рида — Соломона. Часть 2 — арифметика полей Галуа»: <https://habr.com/ru/companies/yadro/articles/341506/>.
12. Видео «Коды Хэмминга — Григорий Кабатянский»: <https://www.youtube.com/watch?v=DycYyYBwb9E>.

Статьи про обработку данных

13. Статья «Линейная аппроксимация» — при обработке экспериментальных данных часто возникает необходимость аппроксимировать их линейной функцией: <https://prog-cpp.ru/mnk/>.
14. Статья «Аппроксимация функции»: <https://libraryno.ru/3-3-approksimaciya-funkcii-matmodosipkina/>.
15. Статья «Методы сжатия данных»: <https://habr.com/ru/articles/251295/>.
16. Статья «Обзор методов сжатия данных»: <https://www.compression.ru/arctest/descript/methods.htm>.

Материал про автокорреляционные функции

17. Видео «Основы ЦОС: Корреляционная функция»: <https://www.youtube.com/watch?v=SR2ettkhmio>.

Курсы и статьи по языкам программирования

18. «Программирование на Python» — достаточная база, особое внимание урокам 3.8 и 3.9: <https://stepik.org/course/67/promo>.
19. «Программирование на Python для решения олимпиадных задач» — наиболее сбалансирован по глубине, особое внимание третьему модулю: <https://stepik.org/course/66634/promo>.
20. «Python: основы и применение» — затрагивает некоторые глубокие особенности языка, но нет уроков по библиотекам обработки данных: <https://stepik.org/course/512/promo>.
21. Курс «Программирование на языке C++ для решения олимпиадных задач»: <https://stepik.org/course/66646/promo>.
22. Статья «Форматирование чисел и текста в Java»: <https://study-java.ru/uroki-java/formatirovanie-chisel-i-texta-v-java/>.

5. Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи предметного тура по двум предметам: информатике и математике 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 48,0 баллов, независимо от уровня.

Заключительный этап

Индивидуальный предметный тур

- информатика — максимально возможный балл за все задачи — 100 баллов;
- математика — 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 (информатика) с весом $K_1 = 0,2$, по предмету

2 (математика) с весом $K_2 = 0,2$, баллы за командное решение задач инженерного тура с весом $K_3 = 0,6$.

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 информатике (предметный тур) ($S_{1 \text{ макс}} = 100$);

S_2 — балл первой части заключительного этапа по математике (предметный тур) ($S_{2 \text{ макс}} = 100$);

S_3 — итоговый балл инженерного командного тура ($S_{3 \text{ макс}} = 100$).

Итого максимально возможный индивидуальный балл участника заключительного этапа — 100 баллов.

Критерий определения победителей и призеров

Чтобы определить победителей и призеров (независимо от класса) на основе индивидуальных результатов участников, был сформирован общий рейтинг всех участников заключительного этапа. С начала рейтинга были выбраны 4 победителя и 8 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	46,34 и выше
Призеры	От 40,74 до 45,21

6. Работа наставника после НТО

Участие школьника в Олимпиаде может завершиться после любого из этапов: первого или второго отборочных, либо после заключительного этапа. В каждом случае после завершения участия наставнику необходимо провести с учениками рефлексию — обсудить полученный опыт и проанализировать, что позволило достичь успеха, а что привело к неудаче. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: <https://academy.sk.ru/events/310>.

Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (Р2Р). Они могут выявить внутренние проблемы команды, для решения которых в план подготовки можно добавить мероприятия, направленные на ее сплочение.

Стоит рассказать, что в истории НТО было много примеров, когда не победив в первый раз, на следующий год участники показывали впечатляющие результаты, одержав победу сразу в нескольких профилях. Конечно, важно отметить, что так происходит только при учете прошлых ошибок и подготовке к Олимпиаде в течение года.

Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с ними помогает наставнику продемонстрировать важность компетенций, развиваемых в процессе участия в НТО, для будущего образования и карьеры школьников. Поддержка родителей помогает мотивировать участников и позволяет выделить необходимое время на занятия в кружке.

С участниками-выпускниками наставнику рекомендуется обсудить их дальнейшее профессиональное развитие и его связь с выбранными профилями НТО. Отдельно можно обратить внимание на льготы для победителей и призеров, предлагаемые в вузах с интересующими ученика направлениями. Кроме того, ряд вузов предлагает льготы для всех финалистов НТО, а также учитывает результаты Конкурса цифровых портфолио «Талант НТО».