



**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДОМ ДЕТСКОГО ТВОРЧЕСТВА «НА 9-ОЙ ЛИНИИ»
ВАСИЛЕОСТРОВСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

Программа **принята**
на педагогическом совете

протокол № 3

от «30» мая 2025 г

Утверждена

Директор ГБУ ДО
ДДТ «На 9-ой линии»

_____ И.В.Петерсон

Приказ № 74

от «16» июня 2025 г

**ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Олимпиадное программирование»

Возраст обучающихся: 12 - 17 лет

Срок реализации: 1 год

Разработчик:
Лубченков Леонид Константинович,
педагог дополнительного образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Адресат программы: школьники 12-17 лет, заинтересованные в программировании и информационных технологиях.

Актуальность программы Современный мир стремительно развивается в сторону цифровизации и внедрения информационных технологий во все сферы жизни. Навыки программирования, алгоритмического мышления и умения решать нестандартные задачи становятся не только востребованными, но и необходимыми для успешного будущего. Образовательная программа "Олимпиадное программирование" ориентирована на учащихся 12-17 лет и предоставляет возможность развить ключевые компетенции, которые будут полезны независимо от выбора будущей профессии.

Преимущества программы для всех участников Программа "Олимпиадное программирование" полезна не только для детей, которые стремятся участвовать в олимпиадах и добиваться побед. Она обладает широким спектром преимуществ, которые оказывают положительное влияние на интеллектуальное и личностное развитие учащихся, независимо от их дальнейших планов:

1. **Развитие логического и критического мышления.** Участие в решении задач повышает способность к анализу, структурированию и логическому построению рассуждений, что полезно в любых учебных и жизненных ситуациях.
2. **Формирование устойчивых навыков решения сложных проблем.** Олимпиадные задачи учат разбивать проблему на этапы и находить нестандартные пути решения, что способствует развитию настойчивости и креативности.
3. **Приобретение опыта работы с современными технологиями.** Даже базовые знания программирования открывают путь к освоению более сложных ИТ-навыков и позволяют применять их для учебных проектов и повседневных задач.
4. **Усиление аналитических способностей.** Умение анализировать данные, просчитывать сценарии и делать выводы повышает конкурентоспособность учащихся в любой профессиональной области.
5. **Развитие навыков самодисциплины и самоконтроля.** Программирование требует внимания к деталям, терпения и высокой концентрации, что положительно влияет на общий уровень самоорганизации учащихся.
6. **Поступление в вузы и дальнейшая профессиональная карьера.** Опыт изучения программирования и алгоритмов предоставляет значительное преимущество при поступлении на технические и инженерные специальности, а также создает базу для освоения перспективных ИТ-профессий.

Уровень освоения: базовый

Объем освоения: 216 часов. **Срок освоения:** 1 год.

Цель программы Формирование у учащихся устойчивых знаний и навыков в области программирования и алгоритмического мышления, а также развитие интеллектуального потенциала, творческих способностей и личностных качеств, способствующих успешной адаптации в современных условиях цифрового мира, и позволяющих результативно участвовать в специализированных конкурсах и олимпиадах.

Задачи программы

Обучающие задачи:

1. Обучить основам программирования на языках C++, Python и других востребованных языках.
2. Ознакомить учащихся с базовыми и продвинутыми алгоритмами и структурами данных.
3. Развить умение решать олимпиадные задачи разного уровня сложности.
4. Сформировать навыки анализа и оптимизации алгоритмов.

Развивающие задачи:

1. Развить логическое и алгоритмическое мышление.
2. Развить аналитические способности и критическое мышление.
3. Формировать креативный подход к решению сложных и нестандартных задач.
4. Развить навыки самодисциплины, настойчивости и самостоятельной работы.

Воспитательные задачи:

1. Воспитать у учащихся терпение, усидчивость и ответственность за результаты своей работы.

2. Сформировать умение работать в команде и эффективно взаимодействовать с другими участниками процесса.
3. Воспитать мотивацию к постоянному саморазвитию и обучению.
4. Развить стремление к достижению поставленных целей и преодолению трудностей.

Планируемые результаты

Предметные результаты:

1. Овладение основами программирования на языках Python и C++.
2. Умение решать задачи разного уровня сложности, используя алгоритмы и структуры данных.
3. Развитие навыков оптимизации программ и анализа алгоритмов.
4. Освоение базовых и продвинутых алгоритмов (жадные алгоритмы, динамическое программирование, графы и т.д.).

Личностные результаты:

1. Формирование ответственности, усидчивости и терпения при выполнении задач.
2. Развитие мотивации к самообразованию и постоянному развитию.
3. Воспитание настойчивости и умения преодолевать трудности.
4. Развитие уверенности в себе и своих силах.

Метапредметные результаты:

1. Развитие логического, алгоритмического и критического мышления.
2. Умение анализировать задачи и разрабатывать оптимальные решения.
3. Формирование навыков командной работы и коммуникации.
4. Развитие навыков самоорганизации и управления временем при выполнении сложных задач.

Язык реализации: государственный язык Российской Федерации (русский язык).

Форма обучения: очная.

Программа предусматривает возможность использования при обучении дистанционных образовательных технологий.

Интернет-ресурсы, платформы и сервисы для реализации программы

Сервисы для проведения видеоконференций:

Платформы для онлайн обучения:

- Stepik
- Hexlet.io
- HTMLAcademy.ru
- Pythontutor.ru
- АСМР.ru
- Информатикс
- Freecodecamp.org

Мессенджеры и социальные сети:

- VK

- WhatsApp

Облачные сервисы:

- Google drive

- Google docs

Онлайн тестирование:

- Google form

Среды программирования и конструирования:

- Repl.it

- Jupyter Notebook

Условия набора на обучение: принимаются обучающиеся 12-17 лет без специальной подготовки.

Формы организации и проведения занятий: аудиторные

1. **Теоретические занятия**, направленные на изучение основ программирования и алгоритмов.
2. **Практические занятия и разбор задач**, включающие индивидуальную и групповую работу над заданиями разного уровня сложности.
3. **Подготовка к олимпиадам**, включающая моделирование условий соревнований и отработку стратегий выполнения задач.
4. **Наглядные игровые занятия**, направленные на лучшее понимание алгоритмов и структур данных. Например, задачи на жадные алгоритмы решаются с использованием конструктора Лего, а задачи на сортировку чисел демонстрируются с помощью табличек с цифрами, которые учащиеся раскладывают на столе.
5. **Анализ и разбор ошибок**, способствующие повышению уровня самостоятельности и уверенности учащихся.

Материально-техническое оснащение

Для реализации данной программы требуется компьютерный класс, оснащенный следующим оборудованием:

- Сервер: компьютеры (рабочие станции), объединенные в локальную сеть и подключенные к ресурсам Интернет;
- Источник бесперебойного питания.
- Сетевое оборудование (концентратор, сетевой кабель (витая пара 5 категории), розетки (5 категории)).
- Оборудование для подключения к ресурсам Интернет (выделенный канал подключения, модем).

Рабочие станции должны иметь следующую конфигурацию:

	Минимальная	Рекомендуемая
Процессор	Частота не менее 1.8ГГц	Частота не менее 1.8ГГц
Оперативная память	1024Мб	2Гб
Жесткий диск	200Гб	320Гб
Видеокарта	256Мб встроенная	512Мб не встроенная
Сетевая карта	Пропускная способность 100Мбит	Пропускная способность 100Мбит
Монитор	Диагональ 17 дюймов	Диагональ 17 дюймов

Сервер должен иметь следующую конфигурацию:

	Минимальная	Рекомендуемая
Процессор	2 шт. с частотой не менее 2,2ГГц	2 шт. с частотой не менее 2,2ГГц
Оперативная память	1024Мб	4Гб
Жесткий диск	2шт. по 320Гб SATA, RAID0	4шт. по 500Гб SATA, RAID0+1
Видеокарта	512Мб не встроенная	1024Мб не встроенная
Сетевая карта	Пропускная способность 100Мбит	Пропускная способность 100Мбит
Монитор	Диагональ 17 дюймов	Диагональ 17 дюймов
Прочее	Привод DVD-RW	Привод DVD-RW

Требуемое программное обеспечение:

Операционное

MS Windows 7 / MS Windows 8 / MS Windows 10 / Linux Ubuntu

Прикладное

Веб-браузер

Python 3.8

PyCharm IDE

Git

Специальное

Редактор кода

Терминал

Архиватор

Файловый менеджер

УЧЕБНЫЙ ПЛАН (216 Ч.)

№ п/п	Наименование раздела, темы	Количество часов			Форма контроля	Самостоятельная работа с использованием дистанционных образовательных технологий
		Всего	Теория	Практика		
1	Экспресс-курс Python	42	13	29	Программа «Генератор уникальных рецептов»	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
2	Примеры задач	81	24	57	Программа «Оптимальная покупка акций»	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
3	Классические подходы к решению задач	54	15	39	Программа «Подъем по лестнице»	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
4	Знакомство с системами автоматической проверки заданий	39	7,5	31,5	Решение задач из отборочного тура Санкт-Петербургской олимпиады по программированию 2023	

					года	
		Всего	216	59,5	156,5	

Утверждена
Директор ГБУ ДО
ДДТ «На 9-ой линии»

_____ И.В.Петерсон
Приказ № 74
от «16» июня 2025 г

**КАЛЕНДАРНЫЙ ГРАФИК РЕАЛИЗАЦИИ
ДООП «ПРОГРАММИРОВАНИЕ. ПОДГОТОВКА К ИТ-ПРОФЕССИИ»
НА 2025-2026 УЧ. ГОД**

Год обучения	Дата начала обучения по программе	Дата окончания обучения по программе	Всего учебных недель	Всего учебных дней	Количество учебных часов	Режим занятия
1 группа	03.09.2025	23.05.2026	36	72	216	2 раза в неделю по 3 часа

**Рабочая программа
К ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ
«Олимпиадное программирование»**

СОДЕРЖАНИЕ ОБУЧЕНИЯ

1. Экспресс-курс Python

Теория: Основные концепции языка, установка и настройка окружения. Обзор основных типов данных (int, float, str, bool). Операторы if, else, elif. Циклы for и while. Определение и вызов функций, параметры и возвращаемые значения. Основные операции со списками и кортежами. Понятие структуры данных. Словари. Исключения и их обработка. Импортирование модулей, работа с библиотеками. Классы, объекты, наследование. Методы строк, форматирование. Чтение и запись файлов в Python. Понимание списковых включений для создания списков. Основные операции над множествами (объединение, пересечение).

Практика: Установка Python и редактора кода, написание первой программы. Создание переменных и работа с ними. Написание программы с использованием условий. Реализация задач с использованием циклов. Создание собственных функций. Работа с коллекциями данных. Использование словарей для хранения данных. Операции над словарями. Итерация по ключам словаря. Написание программы с обработкой ошибок. Использование стандартных библиотек Python. Создание простого класса. Решение задач на манипуляцию строками. Создание программы для работы с текстовыми файлами. Применение списковых включений для обработки данных. Решение задач с использованием множеств.

Формы контроля: Программа «Генератор уникальных рецептов»

2. Примеры задач

Теория: Определение оптимизационных задач, примеры. Основы комбинаторики, перестановки, сочетания. Основные понятия теории графов. определение и примеры задач: подсчёт количества путей и циклов. типовые задачи с графами из ОГЭ по информатике. построение случайного лабиринта и поиск пути в нем. основы двоичной системы счисления. понятие битовой операции и реализация битовых операций в Python. основные операции над строками. наивный метод поиска подстроки и его применимость. основы шифрования и применения алгоритмов в безопасности данных. использование математических формул в программировании. основы геометрии в контексте программирования. Как подсчитать количество вхождений элементов в массив. Работа с двумерными массивами. История и принципы работы клеточного автомата. Как простые правила приводят к сложному поведению. Введение в динамическое программирование. Принципы жадных алгоритмов, примеры. Основы рекурсии, примеры использования. Условие задачи и её связь с рекурсией. Основные понятия теории чисел (простые числа, делимость). Алгоритм Евклида и его применение для нахождения НОД. Принципы разбиения чисел на суммы. Алгоритмы поиска подстрок (например, КМР). Определение максимальных подмножеств в массивах. Принципы парного соответствия (например, задача о стабильных браках). Физика в программировании

Практика: Решение простой оптимизационной задачи. Решение задач на комбинаторику. Решение задач на поиск кратчайшего пути. написание программы для подсчёта количества путей между двумя вершинами. решение типовых задач. реализация поиска пути в лабиринте. преобразование чисел между системами счисления. решение задач с использованием битовых манипуляций. решение задач на манипуляцию строками. реализация программы, которая ищет все вхождения подстроки в строку. написание программы для шифрования и дешифрования текста с использованием шифра Цезаря.

решение задач с применением формул. решение геометрических задач с помощью кода. Реализация программы для вывода всех чисел массива и их частот. Реализация алгоритма нахождения суммы элементов в подматрице. Написание программы для симуляции "Жизни" Конвея с визуализацией. Решение классической задачи о рюкзаке. Реализация жадного алгоритма для решения задачи. Написание рекурсивной функции для вычисления факториала. Реализация программы, которая решает задачу Ханойской башни и выводит пошаговое решение. реализация функции для проверки, является ли число простым; вывод первых N простых чисел. Реализация алгоритма Евклида для двух и нескольких чисел. Реализация алгоритма разбиения числа на слагаемые. Реализация алгоритма поиска подстроки в строке. Решение задачи о максимальной подпоследовательности. Реализация алгоритма для нахождения парного соответствия. решение задачи о поиске наиболее быстрого пути; задача о метании снаряда под оптимальным углом

Формы контроля: Программа «Оптимальная покупка акций»

3. Классические подходы к решению задач

Теория: Определение метода брутфорса и его применение. Характеристика жадных алгоритмов и их применение в задачах. Преимущества и ограничения жадных стратегий. Примеры применения в задачах оптимизации. Принципы динамического программирования и его преимущества. Понятие рекурсии и её применение в различных задачах. Итеративные и рекурсивные методы генерации подмножеств. Различные методы поиска (линейный, бинарный). Обзор известных алгоритмов сортировки (пузырьковая, вставками). принцип работы алгоритма, сравнение с другими более примитивными алгоритмами сортировки. Подходы к оптимизации алгоритмов. Этапы решения олимпиадных задач (анализ, планирование). Оценка временной сложности алгоритмов (О-нотация). Основы вероятностных методов. Использование случайных чисел для приближённых вычислений. Применение случайного блуждания в моделировании и статистике. Алгоритмы обхода графов (DFS, BFS). Что такое деревья: определение, основные термины (корень, узлы, листья, высота дерева и т. д.). Основные виды деревьев (бинарные, сбалансированные, поисковые деревья и др.). Зачем нужны деревья: примеры применения (поиск, сортировка, хранение иерархий). Алгоритмы Краскала и Прима для нахождения MST.

Практика: Реализация брутфорс-решения для задачи поиска максимума. Реализация простого метода перебора для комбинаторной задачи. Решение задачи о сдаче монетами с использованием жадного алгоритма. Решение задачи о максимальном покрытии интервалов с использованием жадного подхода. Решение задачи о нахождении длинной общей подпоследовательности. Написание программы для решения задачи Фибоначчи рекурсивно. Реализация программы, которая генерирует все подмножества заданного множества. Реализация бинарного поиска в массиве. Реализация сортировки массива различными методами. реализация сортировки слиянием и оценка скорости работы алгоритма. Оптимизация ранее написанного кода для повышения производительности. Применение стратегий на примере реальных олимпиадных задач. Анализ временной сложности различных решений задач. Решение вероятностных задач. Написание программы, которая вычисляет число Пи с использованием метода Монте-Карло. Написание программы для визуализации случайного блуждания точки на плоскости. Реализация обхода графа в глубину или ширину. Реализация базовых операций с деревьями (вставка, удаление, обход в глубину и ширину). Построение и визуализация простого бинарного дерева. Реализация одного из алгоритмов для нахождения MST.

Формы контроля: Программа «Подъем по лестнице»

4. Знакомство с системами автоматической проверки заданий

Теория: Зачем нужны системы автоматической проверки заданий? Структура тестовых наборов для проверки решений. Методы проверки корректности решений. Как интерпретировать результаты проверки? разбор задач из варианта одной из прошедших олимпиад. Формат онлайн-соревнований и правила участия. Как анализировать свои

ошибки после соревнований? Как улучшить производительность своего кода. Зачем нужно портфолио программиста? Основы работы в команде программистов: распределение задач, совместное решение проблем, использование систем контроля версий (Git). Анализ изученных тем и их практического применения в задачах. Оценка сильных и слабых сторон: что удалось понять, а где требуется больше практики. Обзор ресурсов для самостоятельного обучения (онлайн-курсы, книги, тренажеры)

Практика: Ознакомление с интерфейсом системы проверки (например, Codeforces). Создание собственного тестового набора для задачи. Проверка своих решений через систему автоматической проверки. Анализ ошибок своих решений по результатам проверки. решение учащимися варианта, аналогичного разобранному на предыдущем занятии. Участие в пробном онлайн-соревновании по программированию. Проведение анализа своих ошибок после соревнований. Оптимизация существующего решения для уменьшения времени выполнения. Составление портфолио своих работ. Составление расписания подготовки перед олимпиадой. Разработка группового проекта: реализация функционала через совместную работу над задачами. Составление личного плана развития с учетом текущих навыков. Выбор соревнований для участия. Выполнение контрольной задачи с элементами из изученных тем

Формы контроля: Решение задач из отборочного тура Санкт-Петербургской олимпиады по программированию 2023 года

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

Группа №1

(пятница, суббота)

№ занятия	Наименование раздела, темы (теория и практика)	Содержание (теоретическая и Практика)	Дата проведения занятия		Количество часов			Формы контроля усвоения материала	Самостоятельная работа с использованием дистанционных образовательных технологий
			по плану	фактически	Теория	Практика	Всего		
Модуль 1 Экспресс-курс Python					13	29	42		
1.	Введение в Python	Теория: Основные концепции языка, установка и настройка окружения. Практика: Установка Python и редактора кода, написание первой программы.	03.09. 2025		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
2.	Типы данных и переменные	Теория: Обзор основных типов данных (int, float, str, bool). Практика: Создание переменных и работа с ними.	06.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
3.	Условные конструкции	Теория: Операторы if, else, elif. Практика: Написание программы с использованием условий.	10.09. 2025		1,5	1,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания

									из раздела Экспресс-курс Python
4.	Циклы	Теория: Циклы for и while. Практика: Реализация задач с использованием циклов.	13.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
5.	Функции	Теория: Определение и вызов функций, параметры и возвращаемые значения. Практика: Создание собственных функций.	17.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
6.	Списки и кортежи	Теория: Основные операции со списками и кортежами. Практика: Работа с коллекциями данных.	20.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
7.	Словари	Теория: Понятие структуры данных. Словари Практика: Использование словарей для хранения данных. Операции над словарями. Итерация по ключам словаря.	24.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания

									из раздела Экспресс-курс Python
8.	Обработка ошибок	Теория: Исключения и их обработка. Практика: Написание программы с обработкой ошибок.	27.09. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
9.	Модули и библиотеки	Теория: Импортирование модулей, работа с библиотеками. Практика: Использование стандартных библиотек Python.	01.10. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
10.	Основы объектно-ориентированного программирования	Теория: Классы, объекты, наследование. Практика: Создание простого класса.	04.10. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python

11.	Работа со строками	Теория: Методы строк, форматирование. Практика: Решение задач на манипуляцию строками.	08.10. 2025		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
12.	Работа с файлами	Теория: Чтение и запись файлов в Python. Практика: Создание программы для работы с текстовыми файлами.	11.10. 2025		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
13.	Списковые включения	Теория: Понимание списковых включений для создания списков. Практика: Применение списковых включений для обработки данных.	15.10. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python
14.	Множества и операции над ними	Теория: Основные операции над множествами (объединение, пересечение). Практика: Решение задач с использованием множеств.	18.10. 2025		1	2	3	Программа «Генератор уникальных рецептов»	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Экспресс-курс Python

Модуль 2 Примеры задач					24	57	81	
15.	Оптимизационные задачи	Теория: Определение оптимизационных задач, примеры. Практика: Решение простой оптимизационной задачи.	22.10. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
16.	Комбинаторные задачи	Теория: Основы комбинаторики, перестановки, сочетания. Практика: Решение задач на комбинаторику.	25.10. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
17.	Задачи на графах	Теория: Основные понятия теории графов. Практика: Решение задач на поиск кратчайшего пути.	29.10. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
18.	Комбинаторные задачи на графах	Теория: определение и примеры задач: подсчёт количества путей и циклов Практика: написание программы для подсчёта количества путей между двумя вершинами	01.11. 2024		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания

								из раздела Примеры задач олимпиадного программирования
19.	Задачи на графы из Основного государственного экзамена по информатике	Теория: типовые задачи с графами из ОГЭ по информатике Практика: решение типовых задач	05.11. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
20.	Поиск путей в лабиринте	Теория: построение случайного лабиринта и поиск пути в нем Практика: реализация поиска пути в лабиринте.	08.11. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
21.	Двоичная арифметика	Теория: основы двоичной системы счисления. Практика: преобразование чисел между системами счисления.	12.11. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
22.	Задачи на битовые операции	Теория: понятие битовой операции и реализация битовых операций в Python.	15.11.		1	2	3	Изучение материалов на сайте

		Практика: решение задач с использованием битовых манипуляций.	2025						ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
23.	Задачи на строки	Теория: основные операции над строками. Практика: решение задач на манипуляцию строками.	19.11. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
24.	Поиск подстроки в строке	Теория: наивный метод поиска подстроки и его применимость. Практика: реализация программы, которая ищет все вхождения подстроки в строку.	22.11. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
25.	Простой алгоритм шифрования (шифр Цезаря)	Теория: основы шифрования и применения алгоритмов в безопасности данных. Практика: написание программы для шифрования и дешифрования текста с использованием шифра Цезаря.	26.11. 2025		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного

								программирования
26.	Простые задачи на формулы	Теория: использование математических формул в программировании. Практика: решение задач с применением формул.	29.11. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
27.	Геометрические задачи	Теория: основы геометрии в контексте программирования. Практика: решение геометрических задач с помощью кода.	03.12. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
28.	Частотный анализ массива	Теория: Как подсчитать количество вхождений элементов в массив. Практика: Реализация программы для вывода всех чисел массива и их частот.	06.12. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
29.	Задачи на матрицы	Теория: Работа с двумерными массивами. Практика: Реализация алгоритма нахождения суммы элементов в подматрице.	10.12. 2025		0,5	2,5	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и

								выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
30.	Игра "Жизнь" Конвея	Теория: История и принципы работы клеточного автомата. Как простые правила приводят к сложному поведению. Практика: Написание программы для симуляции "Жизни" Конвея с визуализацией.	13.12. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
31.	Задачи на динамическое программирование	Теория: Введение в динамическое программирование. Практика: Решение классической задачи о рюкзаке.	17.12. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
32.	Задачи на жадные алгоритмы	Теория: Принципы жадных алгоритмов, примеры. Практика: Реализация жадного алгоритма для решения задачи.	20.12. 2025		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования

33.	Задачи на рекурсию	Теория: Основы рекурсии, примеры использования. Практика: Написание рекурсивной функции для вычисления факториала.	24.12. 2025		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
34.	Ханойская башня	Теория: Условие задачи и её связь с рекурсией. Практика: Реализация программы, которая решает задачу Ханойской башни и выводит пошаговое решение.	27.12. 2025		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
35.	Задачи на теорию чисел	Теория: Основные понятия теории чисел (простые числа, делимость). Практика: реализация функции для проверки, является ли число простым; вывод первых N простых чисел	10.01. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
36.	Нахождение наибольшего общего делителя (НОД)	Теория: Алгоритм Евклида и его применение для нахождения НОД. Практика: Реализация алгоритма Евклида для двух и нескольких чисел.	14.01. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания

								из раздела Примеры задач олимпиадного программирования
37.	Задачи на разбиение чисел	Теория: Принципы разбиения чисел на суммы. Практика: Реализация алгоритма разбиения числа на слагаемые.	17.01. 2026		0,5	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
38.	Задачи на нахождение подстроки	Теория: Алгоритмы поиска подстрок (например, КМР). Практика: Реализация алгоритма поиска подстроки в строке.	21.01 2026		0,5	2,5	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
39.	Задачи на максимальные подмножества	Теория: Определение максимальных подмножеств в массивах. Практика: Решение задачи о максимальной подпоследовательности.	24.01 2026		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования

40.	Задачи на парное соответствие	Теория: Принципы парного соответствия (например, задача о стабильных браках). Практика: Реализация алгоритма для нахождения парного соответствия	28.01. 2026		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
41.	Задачи с физическим сюжетом	Теория: физика в программировании Практика: решение задачи о поиске наиболее быстрого пути; задача о метании снаряда под оптимальным углом	31.01. 2026		1	2	3	Программа «Оптимальная покупка акций»	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Примеры задач олимпиадного программирования
Модуль 3 Классические подходы к решению задач					15	39	54		
42.	Брутфорс (перебор)	Теория : Определение метода брутфорса и его применение. Обзор методов перебора решений задач Практика : Реализация брутфорс-решения для задачи поиска максимума. Реализация простого метода перебора для комбинаторной задачи	04.02. 2026		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
43.	Жадные алгоритмы	Теория: Характеристика жадных алгоритмов и их применение в задачах Практика: Решение задачи о сдаче монетами с	07.02. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и

		использованием жадного алгоритма							выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
44.	Жадные стратегии в реальных задачах	Теория: Преимущества и ограничения жадных стратегий. Примеры применения в задачах оптимизации. Практика: Решение задачи о максимальном покрытии интервалов с использованием жадного подхода.	11.02. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
45.	Динамическое программирование	Теория: Принципы динамического программирования и его преимущества Практика: Решение задачи о нахождении длинной общей подпоследовательности	14.02. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
46.	Рекурсия	Теория: Понятие рекурсии и её применение в различных задачах Практика: Написание	18.02. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и

		программы для решения задачи Фибоначчи рекурсивно						выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
47.	Генерация всех подмножеств множества	Теория: Итеративные и рекурсивные методы генерации подмножеств. Практика: Реализация программы, которая генерирует все подмножества заданного множества.	21.02. 2026		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
48.	Алгоритмы поиска	Теория: Различные методы поиска (линейный, бинарный) Практика: Реализация бинарного поиска в массиве	25.02. 2026		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
49.	Алгоритмы сортировки	Теория: Обзор известных алгоритмов сортировки (пузырьковая, вставками) Практика: Реализация	28.02. 2026		0,5	2,5	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и

		сортировки массива различными методами							выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
50.	Сортировка слиянием	Теория: принцип работы алгоритма, сравнение с другими более примитивными алгоритмами сортировки Практика: реализация сортировки слиянием и оценка скорости работы алгоритма	04.03. 2026		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
51.	Методы оптимизации решений	Теория: Подходы к оптимизации алгоритмов Практика: Оптимизация ранее написанного кода для повышения производительности	07.03. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
52.	Стратегии решения олимпиадных задач	Теория: Этапы решения олимпиадных задач (анализ, планирование) Практика: Применение	11.03. 2026		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и

		стратегий на примере реальных олимпиадных задач						выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
53.	Работа с временной сложностью алгоритмов	Теория: Оценка временной сложности алгоритмов (O-нотация) Практика: Анализ временной сложности различных решений задач	14.03. 2026		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
54.	Методы вычисления вероятностей	Теория: Основы вероятностных методов Практика: Решение вероятностных задач	18.03. 2026		1	2	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
55.	Число Пи методом Монте-Карло	Теория: Использование случайных чисел для приближённых вычислений. Практика: Написание	21.03. 2026		0,5	2,5	3	Изучение материалов на сайте ddtpython.pythonanywhere.com и

		программы, которая вычисляет число Пи с использованием метода Монте-Карло.							выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
56.	Алгоритмы случайного блуждания (Random Walk)	Теория: Применение случайного блуждания в моделировании и статистике. Практика: Написание программы для визуализации случайного блуждания точки на плоскости.	25.03. 2026		0,5	2,5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
57.	Методы работы с графами	Теория: Алгоритмы обхода графов (DFS, BFS) Практика: Реализация обхода графа в глубину или ширину	28.03. 2026		1	2	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования
58.	Введение в деревья	Теория: Что такое деревья: определение, основные	01.04. 2026		1.5	1.5	3		Изучение материалов на сайте ddtpython.pythonanywhere.com и

		<p>термины (корень, узлы, листья, высота дерева и т. д.)</p> <p>Основные виды деревьев (бинарные, сбалансированные, поисковые деревья и др.)</p> <p>Зачем нужны деревья: примеры применения (поиск, сортировка, хранение иерархий)</p> <p>Практика:</p> <p>Реализация базовых операций с деревьями (вставка, удаление, обход в глубину и ширину)</p> <p>Построение и визуализация простого бинарного дерева</p>							<p>выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного программирования</p>
59.	Задачи на минимальное остовное дерево (MST)	<p>Теория: Алгоритмы Краскала и Прима для нахождения MST</p> <p>Практика: Реализация одного из алгоритмов для нахождения MST</p>	04.04.2026		0,5	1,5	3	Программа «Подъем по лестнице»	<p>Изучение материалов на сайте ddtpython.pythonanywhere.com и выполнение домашнего задания из раздела Классические подходы к решению задач олимпиадного</p>

									программирования
Модуль 4 Знакомство с системами автоматической проверки заданий					7,5	31,5	39		
60.	Введение в системы автоматической проверки	Теория: Зачем нужны системы автоматической проверки заданий? Практика: Ознакомление с интерфейсом системы проверки (например, Codeforces).	08.04. 2026		0,5	2,5	3		
61.	Как работают тестовые наборы?	Теория: Структура тестовых наборов для проверки решений Практика: Создание собственного тестового набора для задачи	11.04. 2026		0,5	2,5	3		
62.	Проверка решений на корректность	Теория: Методы проверки корректности решений Практика: Проверка своих решений через систему автоматической проверки	15.04. 2026		0,5	2,5	3		
63.	Анализ результатов проверки заданий	Теория: Как интерпретировать результаты проверки? Практика: Анализ ошибок своих решений по результатам проверки	18.04. 2026		0,5	2,5	3		
64.	Исчерпывающий разбор варианта с реальной олимпиады по программированию	Теория: разбор задач из варианта одной из прошедших олимпиад	22.04. 2026		2	1	3		
65.	Решение варианта	Практика: решение учащимися варианта, аналогичного разобранным на предыдущем занятии	25.04. 2026		0	3	3	Решение задач из отборочного тура Санкт-Петербургской олимпиады по программированию	

								2023 года	
66.	Участие в онлайн-соревнованиях по программированию	Теория: Формат онлайн-соревнований и правила участия Практика: Участие в пробном онлайн-соревновании по программированию	29.04. 2026		0,5	2,5	3		
67.	Работа над ошибками после соревнований	Теория: Как анализировать свои ошибки после соревнований? Практика: Проведение анализа своих ошибок после соревнований	02.05. 2026		0,5	2,5	3		
68.	Оптимизация кода для автоматической проверки	Теория: Как улучшить производительность своего кода Практика: Оптимизация существующего решения для уменьшения времени выполнения	06.05. 2026		0,5	2,5	3		
69.	Создание портфолио своих работ	Теория: Зачем нужно портфолио программиста? Практика: Составление портфолио своих работ	13.05. 2026		0,5	2,5	3		
70.	Подготовительные мероприятия перед олимпиадой	Теория: Как подготовиться к олимпиаде за короткий срок? Практика: Составление расписания подготовки перед олимпиадой	16.05. 2026		0,5	2,5	3		
71.	Технологии командной разработки	Теория: Основы работы в команде программистов: распределение задач, совместное решение проблем, использование систем контроля версий (Git). Практика: Разработка группового проекта:	20.05. 2026		0,5	2,5	3		

		реализация функционала через совместную работу над задачами.							
72.	Подведение итогов и планирование траектории дальнейшего развития	<p>Теория:</p> <p>Анализ изученных тем и их практического применения в задачах</p> <p>Оценка сильных и слабых сторон: что удалось понять, а где требуется больше практики</p> <p>Обзор ресурсов для самостоятельного обучения (онлайн-курсы, книги, тренажеры)</p> <p>Практика:</p> <p>Составление личного плана развития с учетом текущих навыков</p> <p>Выбор соревнований для участия</p> <p>Выполнение контрольной задачи с элементами из изученных тем</p>	23.05	0,5	2,5	3			
	Всего			59,5	156,5	216			

МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
СИСТЕМА ОЦЕНИВАНИЯ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ

Критерии		Оценка
1.	Код не сдан	0
2.	Код сдан, но не запускается	1
3.	Код запускается, но имеет логическую ошибку, отчего вывод неверный	2
4.	Вывод программы верный, но в программе содержится более трех стилистических ошибок	3
5.	Вывод программы верный, но в программе содержится от 1 до 3 стилистических ошибок	4
6.	Вывод программы верный, и код написан без стилистических ошибок	5

СИСТЕМА НАЧИСЛЕНИЯ БАЛЛОВ ПО РАЗДЕЛАМ ПРОГРАММЫ

Задание	Баллы	
	Мин.	Макс.
Модуль 1. Экспресс-курс Python		
Программа «Генератор уникальных рецептов»	0	5
Модуль 2. Примеры задач		
Программа «Оптимальная покупка акций»	0	5
Модуль 3. Классические подходы к решению задач		
Программа «Подъем по лестнице»	0	5
Модуль 4. Знакомство с системами автоматической проверки заданий		
Решение задач из отборочного тура Санкт-Петербургской олимпиады по программированию 2023 года	0	30
А. Удлинитель	0	5
В. Числа в таблице	0	5
С. Последняя цифра	0	5
Д. Кукушка	0	5
Е. Облака	0	5
Г. Выравнивание сыра	0	5

	ИТОГО	0	45
--	--------------	----------	-----------

ЗАДАНИЯ ДЛЯ ТЕКУЩЕГО И ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Программа «Генератор уникальных рецептов»

Написать программу, которая:

- Читает список ингредиентов из файла (например, ingredients.txt).
- Генерирует случайный рецепт (комбинацию из 3–5 ингредиентов).
- Проверяет, чтобы рецепт не повторялся.
- Записывает созданные рецепты в другой файл (recipes.txt).

Цель: Проверка навыков работы с файлами, списками и генерацией случайных данных.

Программа «Оптимальная покупка акций»

На фондовом рынке есть товар с бесконечным количеством акций. Цены акций заданы на N дней, где $arr[i]$ обозначает цену акции в i -й день. Существует правило, что клиент может купить максимум i акций в i -й день. Если у клиента изначально есть k сумма денег, нужно выяснить, максимальное количество акций, которое он может купить.

Например:

- Для 3 дней, цены акций заданы как 7, 10, 4.
- В 1-й день можно купить 1 акцию за 7 р.
- Во 2-й день можно купить до 2 акций, каждая из которых стоит 10 р.
- В 3-й день можно купить до 3 акций, каждая из которых стоит 4 р.

Программа «Подъем по лестнице»

Есть n ступенек, и человек, стоящий внизу, хочет подняться наверх. Человек может подниматься либо на 1 ступеньку, либо на 2 ступеньки за раз. Задача — подсчитать количество способов, которыми человек может достичь вершины.

Примечание: Эта задача похожа на задачу "Подсчитать количество способов добраться до N -й ступеньки (порядок не важен)", с той разницей, что в данном случае учитываются все различные способы, где разный порядок шагов считается уникальным.

Примеры:

- **Ввод:** $n = 1$
Вывод: 1
Объяснение: Есть только один способ подняться на 1 ступеньку.
- **Ввод:** $n = 2$
Вывод: 2
Объяснение: Есть два способа достичь 2-й ступеньки: $\{1, 1\}$ и $\{2\}$.
- **Ввод:** $n = 4$
Вывод: 5
Объяснение: Есть пять способов достичь 4-й ступеньки: $\{1, 1, 1, 1\}$, $\{1, 1, 2\}$, $\{2, 1, 1\}$, $\{1, 2, 1\}$, и $\{2, 2\}$.

А. Удлинитель

Миша решил позвать в гости друзей, чтобы вместе позаниматься программированием. Однако скоро он заметил проблему: в его комнате всего одна свободная розетка, а ему и его друзьям нужно будет подключить к сети n ноутбуков. Миша нашел в магазине удлинители на 5 розеток. Помогите ему посчитать, какое число удлинителей ему нужно купить, чтобы можно было подключить все n ноутбуков.

Входные данные: Ввод содержит число n

Выходные данные: Выведите одно число — необходимое число удлинителей.

В. Числа в таблице

Вова нарисовал квадратную таблицу $n \times n$ клеток. Он расставил в клетки таблицы числа от 1 до n^2 следующим образом: в первую строку слева направо он расставил числа от 1 до n , во вторую строку слева направо он расставил числа от $n+1$ до $2 \cdot n$, и т. д. Например, если $n=5$, таблица будет выглядеть так:

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

Вам нужно помочь Вове написать программу, которая будет узнавать, какое число находится в i -й строке и j -м столбце полученной таблицы.

Входные данные: Первая, вторая и третья строка содержат числа n , i и j , соответственно ($1 \leq n \leq 50$, $1 \leq i, j \leq n$).

Выходные данные: Выведите число, которое находится в i -й строке и j -м столбце полученной таблицы.

С. Последняя цифра

Однажды Таня узнала про операцию возведения в степень. Она работает следующим образом: a^b означает, что нужно умножить число a само на себя b раз. Например, $2^5 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 32$

Таня заметила, что если увеличивать степень n , то число 2^n растет очень быстро:

$$2^5 = 32$$

$$2^{10} = 1024$$

$$2^{20}=1048576$$

$$2^{50}=1125899906842624$$

Для больших чисел n посчитать число 2^n оказалось очень сложно, поэтому Таня решила упростить задачу. Она хочет найти только последнюю цифру числа 2^n . Помогите ей это сделать.

Входные данные: Ввод содержит одно число n

Выходные данные: Выведите последнюю цифру числа 2^n .

Д. Кукушка

Леша занимается в комнате с кукушкой. Каждый час кукушка кукует: в 01:00 один раз, в 02:00 два раза, ..., в 12:00 двенадцать раз, далее в 13:00 снова один раз, в 14:00 два раза, ..., в 23:00 одиннадцать раз, в 00:00 двенадцать раз. Леша знает, во сколько начинается и заканчивается занятие. Помогите ему посчитать, сколько раз суммарно прокукует кукушка за это время.

Входные данные:

Первая строка содержит время начала занятия в формате НН:ММ, где НН — часы, а ММ — минуты. Вторая строка содержит время конца занятия в том же формате.

Гарантируется, что оба момента времени находятся в диапазоне от 00:01 до 23:59, время начала раньше, чем время конца, и ни в одно из них не кукует кукушка (то есть число минут ММ не равно нулю).

Выходные данные:

Выведите одно число — сколько раз суммарно прокукует кукушка за время занятия.

Е. Облака

Маша любит наблюдать за облаками. Она делит облака на три группы: маленькие, средние и большие. Она заметила, что конфигурация облаков на небе меняется по следующим правилам. За один шаг совершается одно из следующих действий:

1. одно большое облако делится на два средних,
2. одно среднее облако делится на два маленьких,
3. два средних облака объединяются в одно большое,
4. два маленьких облака объединяются в одно среднее.

Маша записала, что утром на небе было a_1 больших облаков, a_2 средних, и a_3 маленьких, а вечером было b_1 больших, b_2 средних, и b_3 маленьких. Помогите Маше понять, какое минимальное число действий могли привести к тому, что из утренней конфигурации получилась вечерняя.

Входные данные:

Первая строка содержит три числа a_1 , a_2 , a_3 . Вторая строка содержит три числа b_1 , b_2 , b_3 . Все числа находятся в диапазоне от 0 до 100.

Выходные данные:

В первой строке выведите одно число k — минимальное число действий, которые могли произойти за день. Во второй строке выведите k чисел — типы действий, в том порядке, в котором они произошли. Если есть несколько ответов, выведите любой. Если ответа нет, выведите одно число -1 .

Г. Выравнивание сыра

Есть n кусков сыра и k мышат. Изначально вес i -го куска сыра равен a_i . Каждую минуту мышата выбирают k кусков сыра, и каждый откусывает от своего по одному грамму. Вам нужно узнать, за какое минимальное число минут мышата могут сделать все куски одинакового ненулевого размера.

Входные данные

Первая строка содержит числа n и k ($1 \leq k \leq n \leq 100$). Вторая строка содержит n чисел a_i ($1 \leq a_i \leq 100$).

Выходные данные

Выведите одно число — минимальное число минут, за которое мышата смогут сделать все куски одинакового размера. Если сделать это невозможно, выведите одно число -1 .

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

1. Перечень образовательных технологий

В процессе обучения и контроля используются следующие образовательные технологии:

1. Информационно-коммуникационные технологии (ИКТ)

- использование языка программирования Python;
- работа с файловой системой;
- применение автоматических систем проверки заданий (online judge).

2. Практико-ориентированное обучение

- решение задач, приближенных к реальным прикладным и олимпиадным ситуациям;
- написание и отладка программ с заданными входными и выходными данными.

3. Модульное обучение

- поэтапное освоение материала (экспресс-курс \rightarrow примеры задач \rightarrow классические алгоритмы \rightarrow олимпиадные задачи);
- накопительная система оценивания.

4. Проблемное обучение

- самостоятельный поиск решений;
- анализ условий задач и выбор оптимальных алгоритмов.

5. Дифференцированное обучение

- задания различного уровня сложности;
- возможность получения частичных баллов.

6. Самостоятельная работа обучающихся

- индивидуальное выполнение программных заданий;
- самостоятельная проверка и исправление ошибок.

2. Перечень методов обучения

В рамках курса используются следующие методы обучения:

1. Объяснительно-иллюстративный метод

- разбор синтаксиса Python;
- демонстрация примеров кода.

2. Практический метод

- написание программ по заданным условиям;
- работа с тестами и входными данными.

3. Метод алгоритмического анализа

- разбор алгоритмов (жадные алгоритмы, динамическое программирование, перебор);
- оценка корректности и эффективности решений.

4. Метод самостоятельного поиска решения

- формирование логического и алгоритмического мышления;
- развитие навыков декомпозиции задач.

5. Контрольно-оценочный метод

- проверка работ по заранее заданным критериям;
- анализ типичных ошибок.

6. Рефлексивный метод

- обсуждение ошибок и способов их исправления;
- сравнение различных подходов к решению задач.

3. Перечень дидактических материалов

3.1. Наглядные материалы

- презентации по темам:
 - основы Python;
 - работа со списками и файлами;
 - случайные величины и генерация данных;
 - динамическое программирование;
- блок-схемы алгоритмов;
- таблицы соответствия условий и алгоритмов.

3.2. Раздаточные материалы

- тексты заданий с примерами входных и выходных данных;
- памятки:
 - основные конструкции Python;
 - типичные ошибки при работе с файлами;
 - требования к стилю кода;
- чек-листы для самопроверки программ.

3.3. Электронные учебные материалы

- шаблоны программ;
- тестовые наборы данных;
- примеры корректных решений;
- автоматические тесты для самоконтроля.

3.4. Контрольно-оценочные материалы

- критерии оценивания программных работ;
- система начисления баллов;
- примеры оцененных работ (эталон, частично верное, с ошибками).

ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

Литература для обучающихся:

1. Доусон М. Програмируем на Python. - СПб.: Питер, 2014. - 416 с.

2. Лутц, М. Изучаем Python, том 1, 5-изд.: Пер.сангл.— СПб.: ООО “Диалектика”, 2019. — 832с.
3. Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. СПб.: Питер, 2018. — 288 с.

Литература для педагога:

1. Шень А. Программирование: теоремы и задачи. 6-е изд., дополненное. М.: МЦНМО, 2017. - 320 с.
2. Голицына О.Л., Партыка Т.Л., Попов И.И. Языки программирования. Учебное пособие. — 2-е изд., перераб. и доп. — М.: Форум, 2010. — 400 с.
3. Цветкова М. С., Великович Л. С. Информатика и ИКТ. М.: Академия,2012. – 352 с.
4. Андреева Е. В. Математические основы информатики. Элективный курс: Учебное пособие. М.: БИНОМ. Лаборатория знаний, 2005 – 328 с.
5. Верещагин Н. К., Шень А. Лекции по математической логике и теории алгоритмов. Часть 2. Языки и исчисления. — 4-е изд., испр. — М.: МЦНМО, 2012. — 240 с

Интернет-ресурсы:

1. <https://ru.hexlet.io/>
2. <https://www.python.org/>
3. <https://pythonworld.ru/>
4. <https://younglinux.info/>
5. <https://hyperskill.org/>
6. <https://stepik.org/>
7. <https://acmp.ru/>
8. <https://informatics.mccme.ru/>
9. <https://www.geeksforgeeks.org/>
10. <https://pep8.ru/>

Интернет источники:

1. Ddtpython – официальный сайт курса со всеми необходимыми материалами для обучения <https://ddtpython.pythonanywhere.com/>
2. Google Teachable Machine – инструмент для обучения школьников основам искусственного интеллекта <https://teachablemachine.withgoogle.com/>
3. Codeforces – платформа для олимпиадного программирования, основанная коллективом разработчиков из университета ИТМО <https://codeforces.com/>
4. GeeksForGeeks – портал с образовательными статьями по различным IT-направлениям <https://www.geeksforgeeks.org/>