

Министерство образования и науки Пермского края  
Государственное бюджетное образовательное учреждение  
«Академия первых»

ПРИНЯТА

педагогическим советом  
ГБОУ «Академия первых»

Протокол от 25.08.22 № 10

УТВЕРЖДЕНА

приказом директора  
ГБОУ «Академия первых»



Ю.В. Трясцина

М.П.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Олимпиадное программирование на C++»

Возраст обучающихся: 13-17 лет

Срок реализации программы: 72 часа.

Составитель программы:  
Перескокова Ольга Ивановна,  
педагог дополнительного  
образования

Пермь  
2022

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**Направленность (профиль) программы.** Настоящая программа «Олимпиадное программирование на C++» является дополнительной общеразвивающей программой технической направленности. Программа относится к базовому уровню реализации, позволяет познакомить учащихся с правилами и особенностями проведения соревнований по олимпиадному программированию, рассмотреть алгоритмы часто встречающиеся в олимпиадах по программированию, а также научить решать некоторые типовые олимпиадные задачи базового уровня сложности.

**Аннотация программы.** Программа рассчитана на обучающихся, которые уже умеют программировать и собираются участвовать в этапах Всероссийской олимпиады школьников и иных олимпиадах по программированию. В рамках реализации настоящей программы рассматриваются не только учебные задачи, но и задачи с настоящих соревнований по программированию, начиная от задач уровня школьного этапа ВСОШ и до несложных задач регионального этапа ВСОШ по информатике. Программа затрагивает алгоритмы, часто встречающиеся в олимпиадных задачах по программированию. Для обучающихся, ранее не программировавших на языке программирования C++, происходит знакомство с языком. Участники, уже владеющих программированием на C++, познакомятся с эффективными методами формирования кода на C++, что поможет уменьшить время написания кода на олимпиадах.

**Актуальность программы.** Происходящая цифровизация экономики и всего современного общества как никогда показала потребность практически всех отраслей человеческой деятельности в квалифицированных кадрах, владеющих навыками и умениями программирования. Приводящиеся как внутри страны, так и за ее пределами турниры и олимпиады по программированию подталкивают детей к постоянному развитию и совершенствованию своих знаний, что и является несомненным преимуществом этой дополнительной общеобразовательной общеразвивающей программы. Программа уникальна в том, что позволяет за краткий срок ее непрерывной интенсивной реализации погрузить ребенка в мир современного продвинутого программирования, его алгоритмов, математических и интеллектуальных моделей.

**Отличительные особенности программы.** Программа «Олимпиадное программирование на C++» рассчитана на интенсивный краткосрочный курс обучения. Она направлена на повышение результативности участников в олимпиадах по программированию. Для этого были выбраны темы, которые встречаются в каждой или почти каждой олимпиаде по программированию.

Преподаватели используют язык программирования C++. Этот язык является наиболее подходящим для участников олимпиад по программированию, т.к. является одним из самых быстрых языков программирования и решения на C++ гарантировано проходят ограничения по времени исполнения.

Отличительной особенностью программы является использование систем автоматической проверки решений. В этом случае значительно снижается время получения «обратной связи» школьником. Важно также, что на ошибку указывает не учитель, а бездушное устройство, поэтому у школьника исчезает страх ошибку совершить. Пользуясь тем, что можно быстро что-нибудь исправить, снова запустить программу и очень быстро узнать, что и этот вариант неверен (или, наоборот, верен), обучаемый довольно активно пробует разные варианты. Тем самым, он учится самостоятельно преодолевать проблемы, искать источники ошибок и т.д. Причем, в отличие от учителя, компьютер может проверить решение очень качественно – за минуту прогнать до сотни разных тестов. Если все они прошли, то программа признается правильной. Если какой-то тест не прошел – то, очевидно, программа содержит ошибку. В такой ситуации ученик учится анализировать задачу, выделять из нее ключевые моменты, понимать, что в его программе может приводить к ошибке.

Использование систем автоматической проверки решений позволяет также создать конкурентную среду и привнести соревновательный компонент в обычные занятия. С их помощью формируются рейтинги, в которых хорошо виден уровень каждого участника программы, и любой из них хорошо понимает, как может улучшить свой рейтинг, дорабатывая задачи пройденных тем.

**Адресат программы.** Программа «Олимпиадное программирование на C++» предназначена для детей 13-17 лет, обучающихся в 8-11 классах общеобразовательных организаций, которые уже освоили начальные и базовые знания в области алгоритмизации и хорошо владеют языком программирования. Набор участников программы осуществляется на основании результатов конкурсного отбора, позволяющего оценить уровень готовности ребенка к обучению. Критерии конкурсного отбора формируются педагогами дополнительного образования, реализующими программу, по согласованию с Экспертным советом ГБОУ «Академия первых (Приложение 1).

**Срок реализации программы:** программа реализуется в течение 72 часов. Программа предполагает интенсивный способ ее реализации за в течение двух недель.

**Формы обучения:** Занятия проводятся в формате очного обучения. По каждой теме учащиеся получают задачи по программированию в виде конкурса.

Состав объединения обучающихся (группы) – 15-20 человек.

**Форма занятий:** Лекции, практические занятия, разбор задач.

## **1. ЦЕЛЬ РЕАЛИЗАЦИИ ПРОГРАММЫ**

**Цель программы.** Целью реализации настоящей дополнительной общеразвивающей программы является совершенствование обучающимися теоретических знаний и практических навыков в области решения олимпиадных задач по программированию базового уровня сложности.

## **ЗАДАЧИ РЕАЛИЗАЦИИ ПРОГРАММЫ**

### **2.1. Образовательные задачи:**

- изучение подходов и алгоритмов, используемых при решении олимпиадных задач базового уровня сложности;
- решение олимпиадных задач по программированию базового уровня сложности;
- совершенствование навыков владения языками программирования;
- знакомство с некоторыми подходами и алгоритмами, используемыми в олимпиадном программировании;
- развитие навыков отладки и тестирования для поиска и исправления ошибок в программе;
- освоение языка программирования C++, как наиболее подходящего для решения олимпиадных задач.

### **2.2. Развивающие задачи:**

- создание условий для личностного развития обучающихся;
- выявление и развитие интеллектуальных и алгоритмических способностей обучающихся;
- повышение мотивации учащихся заниматься саморазвитием в области олимпиадного программирования;
- развитие психологической готовности к участию в олимпиадах по программированию.

### 2.3. Воспитательные задачи:

- развитие навыков коммуникации, умения организованно заниматься в коллективе, организовывать работу в парах и мини-группах;
- обеспечение самостоятельности при решении задач учащимися и формирование негативного отношения ко всем вариантам заимствования чужих решений;
- формирование позитивного отношения к любым специализациям IT-отрасли и нацеленности на дальнейшее развитие в этой области безотносительно к успехам или неудачам в области олимпиадного программирования.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

В результате освоения дополнительной общеразвивающей программы «Олимпиадное программирование на C++» учащийся должен знать: методы оценки сложности алгоритма по времени исполнения, алгоритмы, необходимые для решения олимпиадных задач базового уровня сложности, встроенные структуры данных и алгоритмы языка C++, которые могут быть полезны при решении олимпиадных задач по программированию

Используя эти знания, обучающийся должен уметь: разработать алгоритм и составить программу на языке программирования для решения поставленной задачи; осуществлять тестирование программы и ее доработку до получения полного решения; оценивать эффективность решения по времени исполнения и по объему используемой памяти.

## 4. СОДЕРЖАНИЕ ПРОГРАММЫ

### УЧЕБНЫЙ ПЛАН

#### ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

«Олимпиадное программирование на C++»

№	Наименование раздела (модуля)/ темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория / разбор задач	Практика	
1.	Основные алгоритмы теории чисел	10	4	6	Контест
2.	Структуры данных «Стек», «Очередь» и «Дек».	11	3	8	Контест

3.	Стандартная сортировка. Структуры данных «Множество» и «Словарь».	11	3	8	Контест
4.	Динамическое программирование	12	4	8	Контест
5.	Бинарный поиск	12	4	8	Контест
6.	Теория графов	12	4	8	Контест
7.	Итоговый контроль	4	2	2	Контест
	ИТОГО	72	24	48	

## СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

### **Тема 1. Основные алгоритмы теории чисел (10 часов).**

*Теория:* Определение простоты числа. Разложение числа на простые множители. Основная теорема арифметики. Алгоритм Евклида для вычисления НОД и НОК. Сложность алгоритма по времени исполнения.

*Практика:* Решение задач. Оценка сложности алгоритма по времени исполнения. Реализация основных алгоритмов теории чисел. Повышение эффективности циклических алгоритмов путем полного ухода от цикла (сворачивание по формуле), либо избавления от одного из вложенных циклов.

### **Тема 2. Структуры данных «Стек», «Очередь» и «Дек» (11 часов).**

*Теория:* Стек: принцип работы, реализация с помощью одномерного массива. Очередь и дек. Реализация стека, очереди и дека средствами STL в C++. Определение правильной скобочной последовательности. Вычисление выражения в обратной польской записи.

*Практика:* Решение задач. Моделирование стека, очереди, дека. Использование стека, очереди, дека при решении задач.

### **Тема 3. Стандартная сортировка. Структуры данных «Множество» и «Словарь» (11 часов).**

*Теория:* Стандартная сортировка в C++ (из библиотеки `algorithm`), в том числе с использованием компараторов. Сортировка подсчетом. Префиксные суммы. Структуры `set` и `map` в C++.

*Практика:* Решение задач. Сортировка массивов. Использование префиксных сумм при решении задач. Использование идей сортировки подсчетом при

решении задач. Определение множества уникальных объектов. Сортировка данных при помощи map. Накопление данных в словаре по ключу. Сортировка данных из словаря по сложному ключу.

#### **Тема 4. Динамическое программирование (12 часов).**

*Теория:* Динамическое программирование. Задача о Кузнечике. Задача о Черепашке. Восстановление пути по ответу. Оценка сложности алгоритма по объему используемой памяти.

*Практика:* Решение задач. Решение задач динамического программирования на поиск суммы и максимума/минимума. Восстановление пути достижения максимума/минимума по ответу.

#### **Тема 5. Бинарный поиск (12 часов).**

*Теория:* Бинарный поиск. Поиск левого и правого вхождения элемента. Использование стандартной реализации бинарного поиска (из библиотеки STL). Бинарный поиск по ответу. Бинарный поиск для монотонных функций.

*Практика:* Решение задач. Использование бинарного поиска в упорядоченном массиве. Использование бинарного поиска по ответу, при поиске значения монотонной функции.

#### **Тема 6. Теория графов (12 часов).**

*Теория:* Способы представления графа в памяти компьютера. Поиск в ширину (BFS). Поиск в глубину (DFS). Определение количества компонент связности графа. Алгоритм Дейкстры. Алгоритм Флойда. Поиск минимального остовного дерева: алгоритм Прима, алгоритм Крускала.

*Практика:* Решение задач. Моделирование графов. Подсчет и определение оптимальных значений с помощью обходов в ширину и в глубину. Поиск кратчайших путей в графе. Построение минимального остовного дерева.

#### **Тема 7. Итоговый контроль (4 часа).**

*Практика:* Итоговое соревнование по программированию с задачами по материалам всего курса.

**КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**  
**(УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН)**  
**ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ**  
**«Олимпиадное программирование на C++»**

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.	Конкретная дата и время указываются преподавателем в журнале				<b>10, в том числе:</b>	<b>Основные алгоритмы теории чисел.</b>		Практическая работа в форме конкурса
1.1.				Лекция	2	Определение простоты числа. Разложение числа на простые множители. Основная теорема арифметики.	Компьютерный класс	
1.2.				Практикум	4	Решение задач. Оценка сложности алгоритма по времени исполнения. Реализация основных алгоритмов теории чисел.	Компьютерный класс	
1.3				Лекция	2	Алгоритм Евклида для вычисления НОД и НОК. Сложность алгоритма по времени исполнения.	Компьютерный класс	
1.4				Практикум	2	Использование алгоритма Евклида для вычисления НОД и НОК при решении задач.	Компьютерный класс	
2.	Конкретная дата и время указываются преподавателем в журнале				<b>11, в том числе:</b>	<b>Структуры данных «Стек», «Очередь» и «Дек»</b>		Практическая работа в форме конкурса



2.1.		Лекция	2	Стек: принцип работы, реализация с помощью одномерного массива. Очередь и дека. Реализация стека, очереди и дека средствами STL в C++. Определение правильной скобочной последовательности. Вычисление выражения в обратной польской записи.	Компьютерный класс	
2.2.		Практикум	4	Решение задач. Моделирование стека, очереди, дека. Использование стека, очереди, дека при решении задач.	Компьютерный класс	
2.3		Беседа, лекция	1	Обсуждение подходов к решению задач. Способы повышения эффективности решений	Компьютерный класс	
2.4		Самостоятельная работа	4	Повышение эффективности циклических алгоритмов путем полного ухода от цикла (сворачивание по формуле), либо избавления от одного из вложенных циклов.	Компьютерный класс	
3.	Конкретная дата и время указываются преподавателем в журнале		<b>11, в том числе:</b>	<b>Стандартная сортировка. Структуры данных «Множество» и «Словарь»</b>		Практическая работа в форме конкурса
3.1.		Лекция	1	Стандартная сортировка в C++ (из библиотеки algorithm), в том числе с использованием компараторов. Сортировка подсчетом. Префиксные суммы.	Компьютерный класс	

3.2.		Практикум	4	Решение задач. Сортировка массивов. Использование префиксных сумм при решении задач. Использование идей сортировки подсчетом при решении задач.	Компьютерный класс	
3.3		Лекция	2	Структуры set и map в C++.	Компьютерный класс	
3.4		Практикум	4	Определение множества уникальных объектов. Сортировка данных при помощи map. Накопление данных в словаре по ключу. Сортировка данных из словаря по сложному ключу.	Компьютерный класс	
4.	Конкретная дата и время указываются преподавателем в журнале		<b>12, в том числе:</b>	<b>Динамическое программирование</b>		Практическая работа в форме конкурса
4.1.		Лекция	2	Задачи динамического программирования: одномерный и двумерный случай. Задача о Кузнечике. Задача о Черепашке.	Компьютерный класс	
4.2.		Практикум	4	Решение задач. Решение задач динамического программирования на поиск суммы и максимума/минимума.	Компьютерный класс	
4.3		Лекция	2	Восстановление пути по ответу. Оценка сложности алгоритма по объему используемой памяти.	Компьютерный класс	
4.4		Практикум	4	Решение задач. Восстановление пути достижения максимума / минимума по ответу.	Компьютерный класс	

5	Конкретная дата и время указываются преподавателем в журнале		<b>12, в том числе:</b>	<b>Бинарный поиск</b>		Практическая работа в форме конкурса
5.1		Лекция	2	Бинарный поиск. Поиск левого и правого вхождения элемента. Использование стандартной реализации бинарного поиска (из библиотеки STL).	Компьютерный класс	
5.2		Практикум	4	Решение задач. Использование бинарного поиска в упорядоченном массиве.	Компьютерный класс	
5.3		Лекция	2	Бинарный поиск по ответу. Бинарный поиск для монотонных функций.	Компьютерный класс	
5.4		Практикум	4	Использование бинарного поиска по ответу, при поиске значения монотонной функции.	Компьютерный класс	
6.	Конкретная дата и время указываются преподавателем в журнале		<b>12, в том числе:</b>	<b>Теория графов</b>		Практическая работа в форме конкурса
6.1.		Лекция	2	Способы представления графа в памяти компьютера. Поиск в ширину (BFS). Поиск в глубину (DFS). Определение количества компонент связности графа.	Компьютерный класс	
6.2.		Практикум	4	Решение задач. Моделирование графов. Подсчет и определение оптимальных значений с помощью обходов в ширину и в глубину.	Компьютерный класс	
6.3		Лекция	2	Алгоритм Дейкстры. Алгоритм Флойда. Поиск минимального остовного дерева: алгоритм Прима, алгоритм Крускала.	Компьютерный класс	

6.4		Практикум	4	Поиск кратчайших путей в графе. Построение минимального остовного дерева.	Компьютерный класс	
7	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Итоговый контроль</b>		Практическая работа в форме конкурса
7.1		Практикум	2	Соревнование по программированию с задачами по всему курсу.	Компьютерный класс	
7.2		Беседа, лекция	1	Разбор задач итогового конкурса	Компьютерный класс	
7.3		Итоговое занятие	1	Подведение итогов. Обсуждение вариантов дальнейшего развития.	Аудитория	

## 5. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ(ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ)

### Материально-технические условия реализации программы

Наименование специализированных помещений, площадок, аудиторий, кабинетов, лабораторий	Форма (вид) занятий	Оборудование, программное обеспечение
Учебный класс (аудитория) при проведении очного обучения.	Лекция, практикум.	Компьютер, мультимедийный проектор, экран, доска, принтер. Каждому обучающемуся предоставляется компьютер или ноутбук для выполнения практических заданий. Для всех компьютеров должен быть доступ в Интернет и предустановленные системы программирования.

Для доступа в информационно- телекоммуникационную сеть Интернет рекомендуется использовать скорость подключения не менее 10 Мбит/сек.

Рабочее место преподавателя и обучающегося для очных занятий оборудуется персональным компьютером или ноутбуком. Компьютеры участников должны обладать следующими характеристиками:

- процессор с частотой не менее 1,5 ГГц;
- не менее 2 Гб оперативной памяти;
- не менее 1 Гб пространства на диске, доступных участнику для сохранения его файлов.

- монитор размером не менее 13 дюймов, разрешение экрана должно составлять не менее 1024\*768 пикселей.

Список компиляторов и сред разработки, которые должны быть предустановлены на компьютерах преподавателя и обучающихся.

Язык программирования	Компилятор / интерпретатор	Среда разработки
C++	MinGW GNU C++, версия 7.3 или более новая	Code Blocks 17.12 или более новая
C++	Microsoft Visual C++, версия 2015 или более новая	Microsoft Visual Studio Community Edition 2015 или более новая

C++	Для любого установленного компилятора	CLion 2016 или более новая
Python 3	Python 3.6 или более новая	IDLE, Wing IDE, PyCharm 2013.1 Community Edition или более новая
Паскаль	PascalABC.NET 3.7 или более новая	Встроенная
Паскаль	Free Pascal 3.0 или более новая	Встроенная
C	GNU C 7.3 или более новая	Code Blocks 17.12 или более новая
C#	Microsoft Visual C#, версия 2015 или более новая	Microsoft Visual Studio Community Edition 2015 или более новая
Java	Oracle Java JDK 8.0 или более новая	Eclipse JDT, IntelliJ IDEA Community Edition

### **Учебно-методическое обеспечение программы**

Образовательная программа содержит лекционную, методическую и практическую подготовку, большее количество времени уделяется выработке прикладных умений, лекции проводятся в интерактивном формате.

В основу программы положен комплекс педагогических технологий:

- системно-деятельностный подход, обеспечивающий развитие компетенций обучающихся, акцентирующий внимание на вовлечённости и самостоятельной работе слушателей;

- технология развивающего обучения, где слушателям предлагаются задачи, выходящие за пределы их зоны комфорта, ставящие школьников перед необходимостью проявления эвристики, а не повторения рутинизированных действий;

- технология формативного оценивания – на основании обратной связи в процессе научения педагог фиксирует наличие отстающих и общее понимание группой пройденного содержательного блока, в случае неудовлетворительной обратной связи, материал подаётся заново в адаптированном виде.

Программа реализуется с учётом ряда педагогических принципов:

- субъект-субъектные отношения, предполагающие соучастие слушателей и диалог, отсутствие жёстких сценариев большей части занятий;

- дифференцированное обучение, строящееся на концепции минимакса – минимальный уровень должны освоить все, но для лидеров, желающих взять больше знаний и навыков, не ставится верхний («достаточный») предел роста, сохраняется индивидуальный характер обучения;

- смена видов и форматов деятельности: для сохранения концентрации и интереса практикуется чередование периодов практической деятельности и усвоения информации, периоды интеллектуальной мобилизации и разрядки, периоды поточной, групповой и индивидуальной работы, также разнообразятся сами занятия (лекция, анализ материалов, беседа, дискуссия, тренинг, игра и т.д.)

Занятия проводит педагог, имеющий высшее педагогическое образование.

Программа составлена с учетом санитарно-гигиенических требований к порядку проведения занятий и адаптирована к возрастным особенностям обучающихся.

## **СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ**

- 1) Томас Х. Кормен, Алгоритмы. Вводный курс. – Вильямс. 2014. – 208 с.
- 2) Стефан Рэнди Дэвис С++ For Dummies, . – Диалектика. 2018. – 336 с.
- 3) Липпман С.Б, Лажойе Ж, Му Б.Э. Язык программирования С++. Базовый курс. – Вильямс. 2014. – 1120 с.
- 4) Густокашин М. Курс лекций по олимпиадной информатике. [Электронный ресурс] URL: <https://informatics.msk.ru/mod/resource/view.php?id=1381> (Дата обращения: 21.03.2022)
- 5) А. Лааксонен, Олимпиадное программирование. Изучение и улучшение алгоритмов на соревнованиях. – ДМК Пресс. 2020.
- 6) Великович Л.С., Цветкова М.С. Программирование для начинающих. – М. Бином. Лаборатория знаний, 2007. – 287 с.
- 7) Волченков С.Г., Корнилов П.А., Белов Ю.А. и др. Ярославские олимпиады по информатике. Сборник задач с решениями. – М. Бином. Лаборатория знаний, 2010. – 405 с.
- 8) Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы. – М.: Бином. Лаборатория знаний, 2007. – 223 с.
- 9) Кирюхин В.М., Цветкова М.С. Информатика. Программы внеурочной деятельности учащихся по подготовке к Всероссийской олимпиаде школьников: 5 – 11 классы М.: Бином. Лаборатория знаний, 2014.

## Задания конкурсного отбора на дополнительную общеразвивающую программу «Олимпиадное программирование на C++»

### Задача А. Цена товара

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Цена товара обозначена в рублях с точностью до копеек, то есть вещественным числом с двумя цифрами после десятичной точки, например, 10.35. В целочисленных переменных получите и выдайте значения целого числа рублей и целого числа копеек в цене товара.

### Формат входных данных

На вход подается вещественное число  $X$ , в котором ровно два знака после десятичной точки.  $0 < X < 1000$ .

### Формат выходных данных

Выведите сначала количество рублей (целым числом), потом количество копеек (тоже в виде целого числа). Числа выводите в одной строке и разделяйте одним пробелом.

### Пример

стандартный ввод	стандартный вывод
10.35	10 35



## Задача В. Квадратный паркет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На заводе по изготовлению паркета было принято решение изготавливать паркет только квадратной формы. Но заготовки поступают в виде прямоугольных листов древесины. Поэтому, в целях экономии, решено было изготавливать паркет самых разных размеров (самое главное, чтобы он был квадратным). Помогите определять максимальное число паркетных досок, которые можно получить из одного листа заготовки. Известно, что отрезать очередную паркетную доску от общего листа можно только одним прямолинейным разрезом.

Требуется написать программу, которая по длине и ширине прямоугольного листа заготовки вычислит, сколько максимально паркетных досок квадратного размера можно из него получить.

### Формат входных данных

В первой строке записано число  $a$  – длина листа заготовок, а во второй  $b$  – ширина листа,  $1 \leq a, b \leq 10000$ .

### Формат выходных данных

В выходном файле должно находиться единственная строка, в которой должно быть записано единственное целое число – максимальное количество квадратных паркетных досок, которые можно получить из данного листа заготовки.

### Пример

стандартный ввод	стандартный вывод
6 4	3

## Задача С. Разбиение на части

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Натуральное число  $N$  представить в виде суммы  $M$  целых примерно равных частей. Назовём примерно равными части, любые две из которых отличаются не более, чем на 1.

### Формат входных данных

На вход подается две строки: в первой строке находится натуральное число  $N$  ( $N < 30000$ ), а во второй - натуральное число  $M$  ( $M < 20$ ).

### Формат выходных данных

Выведите строку, содержащую  $M$  чисел — частей разбиения числа  $N$  — в порядке неубывания.

### Пример

стандартный ввод	стандартный вывод
13 4	3 3 3 4

## Задача D. Миллионер

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Один миллионер в свой день рождения, когда он заработал свой первый миллион, наклеил на стену 1, и потом каждый год, он в свой день рождения наклеивал на стену число в 10 раз больше предыдущего наклеенного числа и располагал это число правее предыдущего, но вплотную к нему.

После двух дней рождений было наклеено 110, после трех - 110100, после четырех - 1101001000, и т.д.

Требуется написать программу, которая определит  $K$ -ю цифру в последовательности, созданной миллионером, (цифры пронумерованы слева направо, нумерация начинается с 1).

### Формат входных данных

На входе в единственной строке задано натуральное число  $K$  ( $K < 10^9$ ).

### Формат выходных данных

Вывести требуется одну цифру – ту, которая стоит в  $K$ -й позиции в данной последовательности.

### Примеры

стандартный ввод	стандартный вывод
4	1
64	0

## Задача Е. Школьная алгебра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Трёхчлен  $a + bx + y$  от двух переменных  $x$  и  $y$  однозначно определяется свободным членом и коэффициентами  $b$  и  $c$ .

Написать программу, которая по заданным  $a$ ,  $b$  и  $c$  выводит соответствующий трёхчлен, записанный с использованием алгебраических соглашений:

- коэффициент при члене, содержащем переменную, опускается, если его модуль равен единице;
- член, коэффициент при котором равен нулю, опускается;
- знак  $+$  опускается, если он предшествует отрицательному коэффициенту;
- знак  $+$  опускается, если он стоит в начале выражения (так называемый унарный плюс);
- знак умножения между коэффициентом и переменной опускается;
- члены выражения запрещено менять местами;
- пробелы внутри выражения отсутствуют.

### Формат входных данных

На вход подается три строки. В первой строке записано число  $a$ , во второй — число  $b$ , в третьей —  $c$ . ( $a, b, c$  — целые числа, модуль которых меньше 30000)

### Формат выходных данных

Требуется вывести строку, которая содержит трёхчлен, записанный с учетом указанных правил.

### Пример

стандартный ввод	стандартный вывод
0 2 -1	$2x-y$

## Задача F. Удаление пробелов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В строке имеется несколько слов, разделенных одним или несколькими пробелами. Убрать из строки лишние пробелы, в том числе в начале и в конце строки, а слова, стоящие в начале предложения, выдать с заглавной буквы.

### Формат входных данных

Вводится одна строка, в которой встречаются только символы латинского алфавита, пробелы и знаки препинания. Между каждыми двумя словами может быть произвольное число пробелов. Перед первым словом и после последнего слова также могут стоять пробелы.

### Формат выходных данных

Вывести строку, полученную из исходной строки удалением лишних пробелов, в которой каждое предложение записано с заглавной буквы. Перед первым и после последнего слова строки пробелов быть не должно. Между каждой парой рядом стоящих слов должен быть ровно один пробел.

Словом считайте последовательность символов, отличных от пробела, ограниченную пробелами или началом/концом строки.

Предложение может заканчиваться точкой, знаком восклицания или знаком вопроса. Гарантируется, что в тексте есть хотя бы одно предложение.

### Примеры

стандартный ввод	стандартный вывод
Hello. world!	Hello. World!
One. two. three etc.	One. Two. Three etc.