

Задача 1.

Международный стандартный книжный номер (ISBN) – это уникальный идентификатор книжных изданий, необходимый для операций с книгой в торговых сетях и учетном ПО. В современных ISBN используется 13 цифр, но в этой задаче мы рассматриваем 10-значные ISBN старого образца – они действовали до 2007 года. 10-значные ISBN представляют собой строку, где первые 9 символов – цифры, а последний символ – контрольный. Контрольный символ представляет собой сумму первых 9 цифр по модулю 11, причем 10 обозначается символом X. Создайте кэш для поиска цен на книги, представленные в системе с помощью ISBN. Вам нужно реализовать методы поиска, вставки и удаления. Для вытеснения кэша используйте политику наименьшего числа использованных записей (LRU). Если ISBN уже присутствует, метод вставки не должен изменять цену, но должен обновить эту запись как последнюю использованную. Поиск также должен обновить эту запись как последнюю использованную.

Задача 2.

Временные промежутки, в течение которых пользователь занят, выполняя определенные задачи, представлены в его расписании в виде отдельных временных интервалов. В каждый интервал может входить несколько задач. Например:

- С 8 до 15 – работа (с 8 до 10 написание кода, с 10 до 12 совещание и т.д.)
- С 18 до 22 – личные дела (с 18 до 19 ужин, с 19 до 21 просмотр сериалов и т.п.)
- С 23 до 7 часов – сон.

Напишите программу, которая добавляет в список дел новые задачи и обновляет интервалы занятости, отсортированные по времени старта.

Например, если в приведенное выше расписание добавить новую задачу с интервалами (17, 20), обновленные промежутки занятости будут иметь вид: (8, 15), (17, 22), (23, 7).

Задача 3.

Вы работаете в команде веб-платформы с прогнозом погоды. Вы работаете над модулем, который будет выдавать статистическую информацию о температуре за определенный период. На вход подается список из целых чисел. Ваша задача — определить, сколько чисел в этом списке являются положительными, сколько отрицательными и сколько из них равны нулю. Вам нужно вывести не сами числа, а их количество в каждой категории.

Формат ввода

Входные данные состоят из одной строки, в которой чередуются целые числа, разделенные пробелами. Длина списка не превышает 100 элементов.

Формат вывода

Одна строка в формате: «выше нуля: А, ниже нуля: В, равна нулю: С», где А, В и С — количество положительных, отрицательных и нулевых чисел в списке соответственно. Будьте внимательны с пробелами и знаками препинания.

Задача 4

Вы участвуете в разработке программного обеспечения для онлайн-опросов. Необходимо разработать программу для фильтрации результатов опросов, которая оставляет только уникальные значения.

Формат ввода

Строка, содержащая разделённые пробелами слова. Длина строки от 3 до 250 символов.

Формат вывода

Строка, содержащая единственный раз встречавшиеся во входной строке слова, разделенные пробелом, сохраняя порядок. Если таких нет, вернуть “-1” (без кавычек).

Пример 1

Входные данные:

кофе чай кофе чай чай кофе молоко

Выходные данные:

молоко

Пример 2

Входные данные:

красный оранжевый жёлтый зелёный голубой синий фиолетовый

Выходные данные:

красный оранжевый жёлтый зелёный голубой синий фиолетовый

Пример 3

Входные данные:

ой ой ой

Выходные данные:

-1.

Задача 5.

Вам требуется разработать программу, проверяющую, подходит ли день для посадки растений по метеорологическим данным.

Дата подходит для посадки, если:

- средняя температура выше 10 и ниже 25 градусов,
- вероятность дождя не выше 20 процентов.

Формат ввода

Строка, содержащая данные в формате <дата>:<средняя температура>:<вероятность дождя>, <...>:<...>:<...>,...

Строка содержит от 1 до 30 записей, дата – целое число, $0 < x < 32$, температура – целое число, $-20 < 0 < 30$, вероятность дождя – целое число $0 \leq x \leq 100$.

Формат вывода

Строка, содержащая даты, подходящие для посадки, разделённые запятыми и пробелами.

ИЛИ

Строка (без кавычек):

“Подходящих дней нет”

Пример 1

Входные данные:

1:15:10, 2:12:5, 3:8:30, 4:20:15, 5:25:0

Выходные данные:

1, 2, 4

Пример 2

Входные данные:

6:5:50, 7:10:20, 8:12:10, 9:11:25, 10:14:0

Выходные данные:

8, 10

Пример 3

Входные данные:

11:9:0, 12:30:15, 13:1:10, 14:7:5, 15:18:25

Выходные данные:

Подходящих дней нет

Задача 6.

Узнать номер карты или телефон человека иногда можно, просто заглянув ему через плечо. Чтобы обезопасить своих пользователей, разработчики скрывают эти данные, оставляя только последние цифры. Ваша задача — написать функцию, которая преобразует все символы, кроме последних четырех, в «#» (решетку).

Формат ввода

Одна строка произвольных символов, обозначающих, например, номер карты, телефон человека или ключ доступа API. Длина строки — не больше 100 символов. В строке гарантированно отсутствуют пробелы.

Формат вывода

Одна строка, в которой содержатся решётки и четыре цифры или буквы. Если длина строки меньше 4 символов, выведите всё, что было изначально.

Пример 1

Входные данные:

123456789

Выходные данные:

#####6789

Пример 2

Входные данные:

abc

Выходные данные:

abc

Задача 7.

В математическом программном обеспечении требуется реализовать функционал отображения матриц произвольного размера.

Формат ввода

Строка разделённых запятыми чисел, в которой первые два места — числа строк и столбцов в матрице, остальные — числа матрицы построчно. Строка гарантированно содержит нужное количество чисел.

Формат вывода

Строка, содержащая числа матрицы, где числа в пределах строки разделены пробелами, а строки между собой — символами новой строки (округлить числа до тысячных).

Пример 1

Входные данные:

2,4,1.1349,2.6876,3.99999,4.5678,5.8712,6.00001,7.19231,8.123012

Выходные данные:

1.135 2.688 4.0 4.568

5.871 6.0 7.192 8.123

Пример 2

Входные данные:

2,2,2,1.55555,1.66666,2

Выходные данные:

2.0 1.556

1.667 2.0

Задача 8.

Вы разрабатываете компонент для цифровой архивной платформы, которая обрабатывает исторические документы. В базе данных хранятся даты, годы и другие числовые значения, записанные римскими цифрами — как это часто встречается в исторических документах, рукописях и надписях.

Для анализа этих данных, сортировки и построения временных шкал необходимо преобразовывать римские числа в десятичные. Ваша задача — реализовать функцию, которая выполняет такую конвертацию.

Римские числа — непозиционная система записи чисел со следующими правилами.

1. Если меньшее число стоит перед большим, они вычитаются:

IVримская = 410

2. Если большее число стоит перед меньшим, они складываются:
VI римская = 610
3. Из числа не может вычитаться больше, чем разница между предыдущим меньшим числом и им самим, например:
IVX – некорректная запись.

Формат ввода

Строка, состоящая из латинских символов, соответствующим римским цифрам, гарантированно являющаяся корректным числом.

Формат вывода

Строка, содержащая то же число, но записанное десятичными цифрами.

Пример 1

Входные данные:

MCMXXCIV

Выходные данные:

1984

Пример 2

Входные данные:

MCCXXXIV

Выходные данные:

1234.

```
roman_numerals = {  
    'I': 1,  
    'V': 5,  
    'X': 10,  
    'L': 50,  
    'C': 100,  
    'D': 500,  
    'M': 1000  
}
```