



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Задание 1. Тайна синего кристалла

Ограничение по времени: 0,1 секунд.

Ограничение по памяти: 256 МВ

В далёкой галактике существует бесконечная двумерная планета, состоящая из крошечных квадратных участков. На этой планете спрятан древний артефакт — «Синий Кристалл», который обладает удивительной способностью: каждую минуту он распространяет свою энергию на соседние участки, окрашивая их в синий цвет. Легенда гласит, что тот, кто сможет предсказать, как будет распространяться энергия кристалла, получит доступ к его безграничной силе.

Вы — исследователь, который нашёл этот кристалл. Вам дано n минут, чтобы изучить его свойства. Ваша задача — определить, сколько участков планеты будут окрашены в синий цвет через n минут после активации кристалла.

Правила задания:

1. В первую минуту вы активируете кристалл, и один произвольный участок планеты окрашивается в синий цвет.
2. Каждую следующую минуту все соседние (смежные по сторонам) неокрашенные участки также окрашиваются в синий цвет.
3. Ваша задача — вычислить количество синих участков через n минут.

Входные данные:

Одно целое число n $(1 \leq n \leq 10^5)$ — количество минут, которые прошли с момента активации кристалла.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Выходные данные:

Одно целое число — количество участков, окрашенных в синий цвет через n минут.

Пример:

Входные данные: «2»

Выходные данные: «5»

Объяснение:

Минута 1: 1 участок окрашен.

Минута 2: 4 соседних участка окрашены.

Примечание:

Участки, окрашенные на предыдущих минутах, остаются синими и продолжают распространять энергию.

Задача требует оптимального решения, так как n может быть достаточно большим.

Решение:

```
```python
def coloredCells(n: int) -> int:
 return 2 * n * (n - 1) + 1

print(coloredCells(int(input())))
```
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Задание 2. Тайна чисел

Ограничение по времени: 2 секунды.

Ограничение по памяти: 256 МВ.

В далеком будущем, где технологии достигли невероятных высот, человечество обнаружило древние артефакты, содержащие загадочные числовые последовательности. Эти последовательности, как оказалось, обладают скрытой "силой", которую можно вычислить, если правильно их обработать.

Каждая последовательность состоит из чисел. Для каждого начального отрезка последовательности (начиная с первого числа и постепенно увеличивая длину) необходимо вычислить его "силу". Сила отрезка определяется следующим образом:

1. Для каждого числа в отрезке нужно прибавить к нему максимальное значение из всех чисел, которые находятся перед ним или равны ему (включая его само).

2. Сила отрезка — это сумма всех таких измененных чисел.

Ваша задача — написать программу, которая для каждого начального отрезка последовательности вычислит его силу и выведет результат.

Входные данные:

На вход подается массив целых чисел *nums* длиной n $(1 \leq n \leq 10^5)$, где каждый элемент массива удовлетворяет условию: $-10^4 \leq \text{nums}[i] \leq 10^4$.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Формат ввода:

Первая строка содержит целое число n — длину массива.

Вторая строка содержит n целых чисел, разделенных пробелами, — элементы массива $a_{i \bmod m}$.

Выходные данные:

Выведите n целых чисел, разделенных пробелами, — силы префиксов массива $a_{i \bmod m}$ длиной от 1 до n .

Пример:

Входные данные:

...

5

2 3 7 5 10

...

Выходные данные:

...

4 10 24 36 56

...

Примечание:

Гарантируется, что входной массив содержит хотя бы один элемент.

Для решения задачи важно учитывать ограничения по времени и памяти, так как входные данные могут быть большими.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Решение:

```
```python
def findPrefixScore(nums: list[int]) -> list[int]:
 n = len(nums)
 result = [] # Массив для хранения сил отрезков
 current_max = float('-inf') # Текущее максимальное значение
 prefix_sum = 0 # Сумма измененных чисел
 for i in range(n):
 # Обновляем текущее максимальное значение
 current_max = max(current_max, nums[i])
 # Вычисляем измененное число и добавляем его к сумме
 prefix_sum += nums[i] + current_max
 # Сохраняем результат
 result.append(prefix_sum)
 return result

input()
print(*findPrefixScore(list(map(int, input().split()))))
```
```

Задача 3. Проверка валидности sudoku

Судoku — это популярная головоломка, в которой нужно заполнить сетку 9x9 числами от 1 до 9 так, чтобы в каждой строке, каждом столбце и каждом из девяти подквадратов 3x3 не было повторяющихся чисел.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Вам дана частично заполненная сетка sudoku. Ваша задача — проверить, является ли текущее заполнение валидным. Для этого нужно убедиться, что:

1. В каждой строке нет повторяющихся чисел (кроме пустых клеток, обозначенных символом `.`).
2. В каждом столбце нет повторяющихся чисел.
3. В каждом из девяти подквадратов 3x3 нет повторяющихся чисел.

Если все три условия выполняются, сетка считается валидной.

Формат ввода:

9 строк, каждая из которых содержит ровно 9 символов (числа от 1 до 9 или `.`)

Выходные данные:

Выведите `Валидна`, если сетка валидна, и `Не валидна` в противном случае.

Решение:

```
```python
```

```
def is_valid_sudoku(board):
```

```
 # Проверяем строки
```

```
 for row in board:
```

```
 numbers = [x for x in row if x != '.']
```

```
 if len(numbers) != len(set(numbers)):
```

```
 return 'Не валидна'
```

```
 # Проверяем столбцы
```

```
 for col in range(9):
```

```
 numbers = [board[row][col] for row in range(9) if board[row][col] != '.']
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

```
if len(numbers) != len(set(numbers)):
 return 'Не валидна'

Проверяем подквадраты 3x3
for i in range(0, 9, 3):
 for j in range(0, 9, 3):
 numbers = [board[x][y] for x in range(i, i + 3) for y in range(j, j + 3) if
board[x][y] != '.']
 if len(numbers) != len(set(numbers)):
 return 'Не валидна'
 return 'Валидна'

board = [input() for _ in range(9)]
Проверка валидности
result = is_valid_sudoku(board)
print(result)
...
```

#### **Задача 4. Волшебный кошелек алхимика**

**Ограничение по времени: 1 секунда**

**Ограничение по памяти: 256 MB**

Вы — начинающий алхимик, и у вас есть волшебный кошелек, который может создавать монеты различных достоинств. Однако, чтобы создать монету, вам нужно потратить немного магической энергии. Ваша задача — собрать определённую сумму денег, используя минимальное количество монет. Если сумму собрать невозможно, вы должны вернуть `-1`, чтобы предупредить других алхимиков об опасности.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

У вас есть: Монеты различных достоинств (каждую монету можно использовать неограниченное количество раз). Целевая сумма, которую нужно собрать.

Ваша цель — найти минимальное количество монет, необходимое для набора суммы, или определить, что это невозможно.

**Формат ввода данных:**

Первая строка:

Целое число `n` — количество различных достоинств монет.

Формат: `n` (целое число,  $1 \leq n \leq 12$ ).

Вторая строка:

Массив целых чисел `coins`, где каждое число `coins[i]` — достоинство  $i$ -й монеты.

Формат: `coins[0] coins[1] ... coins[n-1]` (целые числа,  $1 \leq coins[i] \leq 2^{31} - 1$ , разделены пробелом).

Третья строка:

Целое число `amount` — сумма, которую нужно собрать.

Формат: `amount` (целое число,  $0 \leq amount \leq 10^4$ ).

**Формат вывода данных:**

Выведите минимальное количество монет, необходимое для набора суммы `amount`. Если сумму собрать невозможно, выведите `-1`.

**Решение:**

```
```python
```

```
def coin_change(coins, amount):
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 1

Инициализируем массив dp, где dp[i] будет хранить минимальное количество монет для суммы i

```
dp = [float('inf')] * (amount + 1)
```

```
dp[0] = 0 # Для суммы 0 монеты не нужны
```

```
# Заполняем массив dp
```

```
for coin in coins:
```

```
    for i in range(coin, amount + 1):
```

```
        dp[i] = min(dp[i], dp[i - coin] + 1)
```

```
# Если сумму нельзя собрать, возвращаем -1, иначе возвращаем dp[amount]
```

```
return dp[amount] if dp[amount] != float('inf') else -1
```

```
n = int(input())
```

```
coins = list(map(int, input().split()))
```

```
amount = int(input())
```

```
print(coin_change(coins, amount))
```

```
...
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

Задача 1. Разлив магической жидкости

Ограничение по времени: 0.1 секунд

Ограничение по памяти: 256 MB

В древнем подземелье алхимики хранили магическую жидкость, обладающую удивительным свойством — она расширяется в строго определённом порядке. Вы нашли старинный сосуд, и теперь ваша задача — вычислить, сколько плиток пола покроется жидкостью через n минут после её разлива.

Правила распространения жидкости:

1. В начале (на 0-й минуте) жидкость уже занимает центральную область размером 2×2 .
2. Каждую минуту жидкость распространяется на все смежные (по сторонам) плитки, которые ещё не покрыты.
3. Ваша задача — вычислить, сколько плиток будет покрыто жидкостью через n минут.

Входные данные:

Целое число n $(1 \leq n \leq 10^5)$ — количество минут после начала разлива.

Выходные данные:

Одно целое число — количество плиток, покрытых жидкостью через n минут.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

Решение:

```
```python
def coloredCells(n: int) -> int:
 return 2 * n 2 + 6 * n + 4
print(coloredCells(int(input())))
```
```

Задача 2. Энергия древнего артефакта

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 MB

Вы — ученый, исследующий древний артефакт, который испускает энергетические импульсы в виде последовательности целых чисел. Ваша задача — найти непрерывную подпоследовательность этих чисел, произведение которых будет максимальным. Это произведение и будет считаться энергией подпоследовательности.

Остерегайтесь отрицательных чисел и нулей! Они могут сильно изменить энергию подпоследовательности.

Входные данные:

На вход подается последовательность целых чисел длиной n $(1 \leq n \leq 2 * 10^4)$, где каждое число находится в диапазоне от -10 до 10.

Формат ввода:

Первая строка содержит целое число n — длину последовательности.

Вторая строка содержит n целых чисел, разделенных пробелами, — энергетические импульсы артефакта.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

Выходные данные:

Выведите одно целое число — максимальную энергию подпоследовательности.

Решение:

```
```python
def max_energy_subsequence(nums):
 if not nums:
 return 0
 # Инициализация переменных
 max_energy = nums[0] # Максимальная энергия
 min_energy = nums[0] # Минимальная энергия (для учета отрицательных чисел)
 result = nums[0] # Результат
 for i in range(1, len(nums)):
 # Если текущее число отрицательное, меняем местами max_energy и min_energy
 if nums[i] < 0:
 max_energy, min_energy = min_energy, max_energy
 # Обновляем max_energy и min_energy для текущего числа
 max_energy = max(nums[i], max_energy * nums[i])
 min_energy = min(nums[i], min_energy * nums[i])
 # Обновляем результат
 result = max(result, max_energy)
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

```
return result

n = int(input()) # Длина последовательности
nums = list(map(int, input().split())) # Энергетические импульсы
Вычисление максимальной энергии
result = max_energy_subsequence(nums)
print(result)
...
```

### Задача 3. Поиск зон заражения

**Ограничение по времени: 1 секунда**

**Ограничение по памяти: 128MB**

В научной лаборатории произошла утечка опасного вируса. Вирус распространяется по лаборатории, которая представлена в виде сетки размером  $m \times n$ . Каждая клетка сетки может быть либо заражена ('1'), либо безопасна ('0'). Вирус распространяется только на соседние клетки по горизонтали и вертикали.

Ваша задача — определить количество отдельных зон заражения. Зона заражения — это группа соседних зараженных клеток, окруженных безопасными клетками или границами сетки.

#### **Входные данные:**

На вход подается сетка размером  $m \times n$ , где каждая клетка содержит либо '1' (заражена), либо '0' (безопасна).



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

### Формат ввода:

Первая строка содержит два целых числа  $m$  и  $n$  ( $1 \leq m, n \leq 300$ ) — размеры сетки. Следующие  $m$  строк содержат по  $n$  символов каждая, представляющих строки сетки. Символы разделены пробелами.

### Выходные данные:

Выведите одно целое число — количество отдельных зон заражения.

### Решение:

```
```python
def num_infected_zones(grid):
    def dfs(i, j):
        # Проверяем границы и безопасность клетки
        if i < 0 or i >= len(grid) or j < 0 or j >= len(grid[0]) or grid[i][j] == '0':
            return
        # Помечаем клетку как безопасную
        grid[i][j] = '0'
        # Рекурсивно проверяем соседние клетки
        dfs(i + 1, j)
        dfs(i - 1, j)
        dfs(i, j + 1)
        dfs(i, j - 1)
    count = 0
    for i in range(len(grid)):
        for j in range(len(grid[0])):
            if grid[i][j] == '1':
```



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

```
# Нашли новую зону заражения
count += 1

# "Очищаем" всю зону с помощью DFS
dfs(i, j)

return count

m, n = map(int, input().split())
grid = [input().split() for _ in range(m)]

# Подсчет зон заражения
result = num_infected_zones(grid)

print(result)

...
```

Задача 4. Тайна волшебных камней

Ограничение по времени: 1 секунда

Ограничение по памяти: 128MB

В далёкой деревне, затерянной среди бескрайних лесов, протекает быстрая и коварная река. Местные жители рассказывают легенду о волшебных камнях, которые появляются на поверхности реки только раз в год. Эти камни обладают магической силой: если встать на один из них, можно совершить прыжок на определённое количество шагов вперёд. Однако, если оступиться и не рассчитать прыжок, можно упасть в воду и не добраться до другого берега.

Ваша задача — помочь жителям деревни определить, можно ли перебраться на другой берег реки, используя магические камни. Каждый камень имеет своё значение, которое указывает максимальное количество



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

шагов, которое можно сделать с этого камня. Вы начинаете с первого камня и должны добраться до последнего.

Формат ввода данных:

Первая строка:

Целое число n — количество камней в реке.

Формат: n (целое число, $1 \leq n \leq 10^4$).

Вторая строка:

Массив целых чисел $nums$, где каждое число $nums[i]$ — значение на i -м камне, указывающее максимальное количество шагов, которое можно сделать с этого камня.

Формат: $nums[0] \ nums[1] \ ... \ nums[n-1]$ (целые числа, $0 \leq nums[i] \leq 10^5$, разделены пробелом).

Формат вывода данных:

Выведите `Да`, если можно добраться до последнего камня, начиная с первого, и `Нет` в противном случае.

Формат: `Да` или `Нет` (без кавычек).

Решение:

```
```python
```

```
def can_jump(nums):
```

```
 max_reach = 0 # Максимальная позиция, до которой можно добраться
```

```
 n = len(nums)
```

```
 for i in range(n):
```

```
 # Если текущая позиция превышает максимальную достижимую позицию,
 # возвращаем False
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 9 класс

Вариант 2

```
if i > max_reach:
 return 'Нет'

Обновляем максимальную достижимую позицию
max_reach = max(max_reach, i + nums[i])

Если максимальная достижимая позиция больше или равна последнему
индексу, возвращаем True
if max_reach >= n - 1:
 return 'Да'
return 'Нет'

Чтение входных данных
n = int(input())
nums = list(map(int, input().split()))

Вывод результата
print(can_jump(nums))
'''
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

### **Задача 1. Антикварный каталог**

**Ограничение по времени: 1 секунды**

**Ограничение по памяти: 128 МВ**

Вы — начинающий коллекционер антиквариата, и недавно вы нашли старинный каталог с описанием редких предметов. Каждый предмет в каталоге имеет свою цену и уровень "красоты" (например, уникальность, историческую ценность или эстетическую привлекательность). Вам нужно помочь своим друзьям, которые хотят купить предметы, но не знают, какой из доступных по их бюджету предмет будет самым красивым.

Ваша задача — для каждого запроса (бюджета) определить, какой предмет с ценой, не превышающей бюджет, обладает максимальной красотой. Если таких предметов нет, то ответом будет 0.

#### **Входные данные:**

Первая строка содержит целое число  $n$  — количество предметов в каталоге.

Следующие  $n$  строк содержат по два целых числа:  $price_i$  и  $beauty_i$ , где  $price_i$  — цена предмета, а  $beauty_i$  — его красота.

Следующая строка содержит целое число  $m$  — количество запросов.

Последние  $m$  строк содержат по одному целому числу  $q_j$  — бюджет для каждого запроса.

$$1 \leq n, m \leq 100000$$

$$1 \leq price_i, beauty_i, q_j \leq 10^9$$



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

**Выходные данные:**

Выведите `m` целых чисел — максимальную красоту предмета, цена которого не превышает бюджет для каждого запроса. Если таких предметов нет, выведите `0`.

**Пример:**

Входные данные:

...

5

1 2

3 2

2 4

5 6

3 5

6

1

2

3

4

5

6

...

Выходные данные:

`2 4 5 5 6 6`



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

**Примечание:**

Предметы могут иметь одинаковую цену, но разную красоту.

**Решение:**

```
```python
def maximumBeauty(items, queries):
    items.sort() # Сортируем по цене
    current_max = 0
    # Обновляем красоту для каждого предмета
    for i in range(len(items)):
        price, beauty = items[i]
        current_max = max(current_max, beauty)
        items[i][1] = current_max
    answer = []
    for q in queries:
        left, right = 0, len(items) - 1
        best = 0
        # Бинарный поиск
        while left <= right:
            mid = (left + right) // 2
            if items[mid][0] <= q:
                best = items[mid][1]
                left = mid + 1
            else:
                right = mid - 1
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

```
answer.append(best)

return answer

items = [list(map(int, input().split())) for _ in range(int(input()))]
queries = [int(input()) for _ in range(int(input()))]
print(*maximumBeauty(items, queries))
...
```

Задача 2. Игра жизнь

Ограничение по времени: 0.5 секунды

Ограничение по памяти: 128 MB

Вам дана сетка размером $m \times n$, представляющая текущее состояние клеточного автомата "Игра Жизни". Каждая клетка может находиться в одном из двух состояний:

Живая (обозначена как `1`).

Мёртвая (обозначена как `0`).

Состояние клетки в следующем поколении определяется на основе её текущего состояния и количества живых соседей. У каждой клетки есть 8 соседей (по горизонтали, вертикали и диагонали).

Правила перехода в следующее поколение:

Для живой клетки:

Если у клетки меньше двух живых соседей, она умирает от "недонаселения".

Если у клетки два или три живых соседа, она остаётся живой.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

Если у клетки больше трёх живых соседей, она умирает от "перенаселения".

Для мёртвой клетки:

Если у клетки ровно три живых соседа, она становится живой ("размножение").

Ваша задача — обновить сетку, чтобы она отражала состояние следующего поколения.

Входные данные:

На вход подается сетка размером $m \times n$, где каждая клетка содержит либо 0, либо 1.

Формат ввода:

Первая строка содержит два целых числа m и n ($1 \leq m, n \leq 25$) — размеры сетки.

Следующие m строк содержат по n символов каждая, разделённых пробелами, представляющих строки сетки.

Выходные данные:

Обновите сетку так, чтобы она отражала состояние следующего поколения. Выведите обновленную сетку в том же формате, что и входные данные (каждая строка сетки выводится через пробел)

Примечание:

Все изменения происходят одновременно. Это означает, что при обновлении состояния клеток нужно использовать исходное состояние сетки, а не уже обновлённые значения.

Сетка может быть любого размера, но не превышает 25x25



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

Решение:

```
```python
def next_generation(grid):
 m, n = len(grid), len(grid[0])
 # Создаем копию сетки для хранения следующего состояния
 next_grid = [[0] * n for _ in range(m)]
 # Направления для поиска соседей (8 соседей)
 directions = [(-1, -1), (-1, 0), (-1, 1),
 (0, -1), (0, 1),
 (1, -1), (1, 0), (1, 1)]
 for i in range(m):
 for j in range(n):
 # Считаем количество живых соседей
 live_neighbors = 0
 for dx, dy in directions:
 x, y = i + dx, j + dy
 if 0 <= x < m and 0 <= y < n and grid[x][y] == 1:
 live_neighbors += 1
 # Применяем правила игры
 if grid[i][j] == 1: # Если клетка живая
 if live_neighbors < 2 or live_neighbors > 3:
 next_grid[i][j] = 0 # Умирает
 else:
 next_grid[i][j] = 1 # Остаётся живой
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

```
 else: # Если клетка мёртвая
 if live_neighbors == 3:
 next_grid[i][j] = 1 # Оживает
Обновляем исходную сетку
for i in range(m):
 for j in range(n):
 grid[i][j] = next_grid[i][j]
m, n = map(int, input().split())
grid = [list(map(int, input().split())) for _ in range(m)]
Вычисление следующего поколения
next_generation(grid)
Вывод обновленной сетки
for row in grid:
 print(" ".join(map(str, row)))
...
```

### Задача 3. Тайна зашифрованного послания

**Ограничение по времени: 1 секунда**

**Ограничение по памяти: 256 MB**

Вы — талантливый криптограф, работающий на секретную организацию. Однажды вам в руки попало загадочное послание, зашифрованное в виде строки цифр.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

Каждая цифра или комбинация цифр соответствует определённой букве русского алфавита:

"1" → 'А'

"2" → 'Б'

...

"32" → 'Я'

Однако, расшифровка послания оказалась не такой простой, как казалось на первый взгляд. Некоторые комбинации цифр можно интерпретировать по-разному. Например, строка "1212" может быть расшифрована как:

"АБАБ" (1, 2, 1, 2)

"ЛБ" (12, 1, 2)

"АВ" (1, 21, 2)

Но комбинация "03" недопустима, так как "0" не соответствует ни одной букве.

Ваша задача — определить, сколько различных способов расшифровки существует для данной строки цифр. Если строка не может быть расшифрована ни одним допустимым способом, верните **'0'**.

#### **Формат ввода данных:**

На вход подаётся строка `s`, состоящая только из цифр.

Формат: `s` (строка длиной от 1 до 100 символов, может содержать ведущие нули).



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

**Формат вывода данных:**

Выведите одно целое число — количество способов расшифровать строку. Если строка не может быть расшифрована, выведите `0`.

Формат: целое число (например, `3` или `0`).

**Решение:**

```
```python
```

```
def num_decodings(s):
```

```
    n = len(s)
```

```
    if n == 0 or s[0] == '0':
```

```
        return 0
```

```
    dp = [0] * (n + 1)
```

```
    dp[0] = 1 # Пустая строка имеет 1 способ расшифровки
```

```
    dp[1] = 1 # Первый символ (если не '0') имеет 1 способ
```

```
    for i in range(2, n + 1):
```

```
        # Проверяем, можно ли использовать текущий символ как однозначное  
        # число
```

```
        if s[i - 1] != '0':
```

```
            dp[i] += dp[i - 1]
```

```
        # Проверяем, можно ли использовать текущий и предыдущий символы  
        # как двузначное число
```

```
        two_digit = int(s[i - 2:i])
```

```
        if 10 <= two_digit <= 32: # Русский алфавит: от 'А' (1) до 'Я' (32)
```

```
            dp[i] += dp[i - 2]
```

```
    return dp[n]
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

Чтение входных данных

```
s = input().strip()
```

Вывод результата

```
print(num_decodings(s))
```

```
...
```

Задача 4. Лягушка в волшебном лесу

Ограничение по времени: 1 секунды

Ограничение по памяти: 128 МВ

На поверхности пруда плавают n кувшинок, пронумерованных от **1** до n . Лягушка начинает своё путешествие с кувшинки под номером **1**. Каждую секунду лягушка прыгает на одну из соседних кувшинок, если она ещё не была посещена. Если доступно несколько кувшинок, лягушка выбирает одну из них случайным образом с равной вероятностью. Если лягушка не может прыгнуть на новую кувшинку, она остаётся на текущей кувшинке навсегда.

Соседние кувшинки соединены мостиками. Мостики задаются в виде массива *bridges*, где каждый элемент — это пара чисел $[a, b]$, обозначающая мостик между кувшинками a и b .

Ваша задача — определить вероятность того, что через t секунд лягушка окажется на кувшинке с номером *target*.

Формат ввода:

Первая строка содержит целое число n ($1 \leq n \leq 100$) — количество вершин в дереве.

Вторая строка содержит целое число m ($m = n - 1$) — количество рёбер в дереве.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

Следующие m строк содержат пары чисел a_i, b_i , разделённых пробелом, — рёбра дерева. ($1 \leq a_i, b_i \leq n$)

Последняя строка содержит два целых числа, разделённых пробелом: $target$ (целевая вершина) и t (количество секунд). ($1 \leq t \leq 50, 1 \leq target \leq n$).

Формат вывода:

Выведите одно вещественное число — вероятность того, что лягушка окажется на вершине $target$ через t секунд. Ответ должен быть точным до 10^{-5} .

Ответ всегда вещественное число.

Примечание:

При $n = 7$, $m = 6$ и при вершинах $[[1,2],[1,3],[1,7],[2,4],[2,6],[3,5]]$, $t = 1$, $target = 7$ можно нарисовать граф.

Таким образом лягушка начинает с вершины 1, прыгая с вероятностью $1/3$, попадает в вершину 7 за 1 секунду.

Решение:

```
```python
```

```
def frog_probability(n, edges, target, t):
```

```
 # Строим граф кувшинок
```

```
 graph = {i: [] for i in range(1, n + 1)} # Создаём словарь для хранения соседей
 каждой кувшинки
```

```
 for a, b in edges:
```

```
 graph[a].append(b) # Добавляем соседей для кувшинки a
```

```
 graph[b].append(a) # Добавляем соседей для кувшинки b
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

# Рекурсивная функция DFS с отслеживанием вероятности

```
def dfs(current, time, parent, probability):
```

```
 """
```

```
 current: текущая кувшинка
```

```
 time: текущее время (количество секунд)
```

```
 parent: предыдущая кувшинка (чтобы не возвращаться назад)
```

```
 probability: текущая вероятность достижения текущей кувшинки
```

```
 """
```

# Если лягушка достигла целевой кувшинки

```
if current == target:
```

```
 # Проверяем, является ли это последним ходом или время истекло
```

```
 if time >= t or len([nei for nei in graph[current] if nei != parent]) == 0:
```

```
 return probability # Возвращаем текущую вероятность
```

```
 return 0.0 # Если можно прыгнуть дальше, но время не истекло,
```

вероятность остаться здесь равна 0

# Если время истекло, но лягушка не на целевой кувшинке

```
if time >= t:
```

```
 return 0.0
```

# Получаем список соседних кувшинок, исключая предыдущую

```
neighbors = [nei for nei in graph[current] if nei != parent]
```

# Если нет соседей, лягушка остаётся на месте

```
if not neighbors:
```

```
 return 1.0 if time == t else 0.0 # Вероятность 1.0, если время истекло,
```

иначе 0.0



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 1

```
Вычисляем вероятность для каждого соседа
 prob = probability / len(neighbors) # Равномерное распределение
вероятности
 return sum(dfs(nei, time + 1, current, prob) for nei in neighbors) # Рекурсивно
вызываем DFS для всех соседей
 # Начинаем с кувшинки 1, время 0, предыдущей кувшинки нет (parent=0),
начальная вероятность 1.0
 return dfs(1, 0, 0, 1.0)
n = int(input()) # Количество кувшинок
m = int(input()) # Количество мостиков
bridges = []
for _ in range(m):
 a, b = map(int, input().split())
 bridges.append((a, b))
target, t = map(int, input().split()) # Целевая кувшинка и время
result = frog_probability(n, bridges, target, t)
print(round(result, 5))
...
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

### **Задача 1. Тайна повторяющегося числа**

**Ограничение по времени: 0.1 секунды**

**Ограничение по памяти: 128 МВ**

В далеком будущем, где человечество колонизировало множество планет, на одной из исследовательских станций произошел сбой в системе управления. Сбой был вызван ошибкой в данных: в последовательности чисел, которая должна была содержать уникальные значения, оказалось одно повторяющееся число.

Ваша задача — найти это число, чтобы устранить сбой и восстановить работу станции. Однако есть два важных условия:

Вы не можете изменять исходную последовательность чисел.

Вы можете использовать только константное количество дополнительной памяти.

#### **Входные данные:**

На вход подается последовательность целых чисел длиной  $n + 1$  ( $1 \leq n \leq 10^5$ ), где каждое число находится в диапазоне от 1 до  $n$ .

#### **Формат ввода:**

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — максимальное значение в последовательности.

Вторая строка содержит  $n + 1$  целых чисел, разделенных пробелами, — саму последовательность.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

**Выходные данные:**

Выведите одно целое число — повторяющееся число в последовательности.

**Решение:**

```
```python
def find_duplicate(nums):
    # Используем алгоритм "черепахи и зайца" для поиска дубликата
    slow = nums[0]
    fast = nums[0]
    # Первый этап: находим точку пересечения
    while True:
        slow = nums[slow]
        fast = nums[nums[fast]]
        if slow == fast:
            break
    # Второй этап: находим начало цикла (дубликат)
    slow = nums[0]
    while slow != fast:
        slow = nums[slow]
        fast = nums[fast]
    return slow

n = int(input()) # Максимальное значение в последовательности
nums = list(map(int, input().split())) # Последовательность чисел
# Поиск повторяющегося числа
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

```
result = find_duplicate(nums)
```

```
print(result)
```

```
...
```

Задача 2. Сапёр

Ограничение по времени: 0.5 секунд

Ограничение по памяти: 128 МВ

Вы играете в игру "Сапёр". Игровое поле представлено в виде сетки размером $m \times n$, где каждая клетка может находиться в одном из следующих состояний:

``M`` — нераскрытая мина.

``E`` — нераскрытая пустая клетка.

``B`` — раскрытая пустая клетка без мин вокруг.

``1`` до ``8`` — раскрытая клетка, указывающая количество мин в соседних клетках.

``X`` — раскрытая мина (игра завершена).

Вам дана текущая конфигурация игрового поля и координаты следующего клика `[click_r, click_c]`. Ваша задача — обновить поле в соответствии с правилами игры:

Если клик приходится на мину ``M``, игра завершается. Мина раскрывается и меняется на ``X``.

Если клик приходится на пустую клетку ``E`` без мин вокруг, она раскрывается и меняется на ``B``. Все соседние нераскрытые клетки (``E``) также раскрываются рекурсивно.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

Если клик приходится на пустую клетку `E` с минами вокруг, она раскрывается и меняется на цифру от `1` до `8`, указывающую количество мин в соседних клетках.

Верните обновлённое игровое поле после выполнения клика.

Входные данные:

На вход подается:

Два целых числа m и n — размеры игрового поля.

Игровое поле в виде m строк, каждая из которых содержит n символов, разделённых пробелами.

Два целых числа $click_r$ и $click_c$ — координаты клика (индексы строки и столбца, начиная с 0).

Формат ввода:

...

m n

<строка 1>

<строка 2>

...

<строка m >

$click_r$ $click_c$

...

Выходные данные:

Выведите обновлённое игровое поле в том же формате, что и входные данные (каждая строка через пробел).



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

Решение:

```
```python
```

```
def update_board(board, click):
```

```
 m, n = len(board), len(board[0])
```

```
 click_r, click_c = click
```

```
 # Если клик на мину, игра завершается
```

```
 if board[click_r][click_c] == 'M':
```

```
 board[click_r][click_c] = 'X'
```

```
 return board
```

```
 # Направления для поиска соседей (8 соседей)
```

```
 directions = [(-1, -1), (-1, 0), (-1, 1),
```

```
 (0, -1), (0, 1),
```

```
 (1, -1), (1, 0), (1, 1)]
```

```
def count_mines(r, c):
```

```
 # Считаем количество мин вокруг клетки (r, c)
```

```
 count = 0
```

```
 for dr, dc in directions:
```

```
 nr, nc = r + dr, c + dc
```

```
 if 0 <= nr < m and 0 <= nc < n and board[nr][nc] == 'M':
```

```
 count += 1
```

```
 return count
```

```
def reveal(r, c):
```

```
 # Если клетка уже раскрыта или это мина, выходим
```

```
 if board[r][c] != 'E':
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

```
 return

Считаем количество мин вокруг
mines = count_mines(r, c)
if mines > 0:
 # Если есть мины, раскрываем клетку как цифру
 board[r][c] = str(mines)
else:
 # Если мин нет, раскрываем как 'B' и рекурсивно раскрываем соседей
 board[r][c] = 'B'
 for dr, dc in directions:
 nr, nc = r + dr, c + dc
 if 0 <= nr < m and 0 <= nc < n:
 reveal(nr, nc)

Раскрываем клетку
reveal(click_r, click_c)
return board

m, n = map(int, input().split())
board = [input().split() for _ in range(m)]
click_r, click_c = map(int, input().split())

Обновление поля
updated_board = update_board(board, [click_r, click_c])

Вывод обновленного поля
for row in updated_board:
 print(" ".join(row))
...
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

### Задача 3. Игра без повторения

**Ограничение по времени: 1 секунда**

**Ограничение по памяти: 256 MB**

Два игрока играют в игру, где они по очереди выбирают числа из пула от 1 до  $n$ . Каждое число можно использовать только один раз. Игрок, который первым достигнет или превысит сумму  $m$ , побеждает. Если первый игрок может гарантированно выиграть при оптимальной игре обоих игроков, верните 1. В противном случае верните 2.

#### Формат ввода данных:

Первая строка:

Целое число  $n$  — максимальное число, которое можно выбрать (от 1 до 20).

Формат:  $n$  (целое число,  $1 \leq n \leq 20$ ).

Вторая строка:

Целое число  $m$  — целевая сумма, которую нужно достичь или превысить.

Формат:  $m$  (целое число,  $0 \leq m \leq 300$ ).

#### Формат вывода данных:

Выведите 1, если первый игрок может гарантированно выиграть, и 2 в противном случае.

#### Решение:

```
```python
```

```
def can_i_win(max_choosable_integer, desired_total):
```

```
    # Если сумма всех чисел меньше desiredTotal, никто не может выиграть
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

```
if (max_choosable_integer * (max_choosable_integer + 1)) // 2 < desired_total:
    return 2

# Если desiredTotal <= 0, первый игрок автоматически выигрывает
if desired_total <= 0:
    return 1

# Используем мемоизацию для хранения уже вычисленных состояний
memo = {}

def dfs(used_numbers, remaining_total):
    # Если состояние уже вычислено, возвращаем результат
    if used_numbers in memo:
        return 1 if memo[used_numbers] else 2

    # Проверяем все возможные ходы
    for i in range(1, max_choosable_integer + 1):
        if not (used_numbers & (1 << i)): # Если число i ещё не использовано
            if remaining_total - i <= 0 or not dfs(used_numbers | (1 << i),
remaining_total - i):
                memo[used_numbers] = True
                return 1

    # Если ни один ход не приводит к победе, возвращаем False
    memo[used_numbers] = 2
    return 2

return dfs(0, desired_total)

max_choosable_integer = int(input())
desired_total = int(input())
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

```
print(can_i_win(max_choosable_integer, desired_total))
```

```
...
```

Задача 4. Мышь и кот

Ограничение по времени: 2.35 секунд

Ограничение по памяти: 256 MB

На неориентированном графе играют два игрока: Мышь и Кот. Они ходят по очереди, начиная с Мыши. Граф задаётся следующим образом: `graph[a]` — это список всех вершин `b`, таких что между `a` и `b` есть ребро.

Мышь начинает игру в вершине 1.

Кот начинает игру в вершине 2.

В вершине 0 находится нора (убежище для Мыши).

На каждом ходу игрок должен переместиться по одному из рёбер графа из текущей вершины. Например, если Мышь находится в вершине 1, она может переместиться в любую вершину из `graph[1]`.

Дополнительные правила:

Кот не может перемещаться в нору (вершину 0).

Игра заканчивается в трёх случаях:

Если Кот оказывается в той же вершине, что и Мышь, Кот побеждает.

Если Мышь достигает норы (вершины 0), Мышь побеждает.

Если позиция повторяется (игроки оказываются в той же ситуации, что и на предыдущем ходу, и очередь хода того же игрока), игра заканчивается ничьей.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

Ваша задача — определить, кто победит при оптимальной игре обоих игроков. Верните:

- 1, если побеждает Мышь,
- 2, если побеждает Кот,
- 0, если игра заканчивается ничьей.

Формат ввода:

Первая строка содержит целое число n — количество вершин в графе ($3 \leq n \leq 50$).

Следующие n строк содержат списки чисел, разделённых пробелами.

Каждая строка i (начиная с 0) описывает вершину i и содержит список вершин, соединённых с ней.

Например, строка $2 \quad 5$ для вершины 0 означает, что вершина 0 соединена с вершинами 2 и 5 .

Количество вершин в каждом списке меньше n .

Вершины нумеруются от 0 до $n-1$.

Вершина 0 — это нора.

Вершины 1 и 2 — стартовые позиции Мыши и Кота соответственно.

Формат вывода:

Выведите одно число:

- 1, если побеждает Мышь,
- 2, если побеждает Кот,
- 0, если игра заканчивается ничьей.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

Пример:

Ввод:

...

4

1 3

0

3

0 2

...

Вывод:

`1

Решение:

```
```python
```

```
def cat_mouse_game(graph):
 from collections import deque
 MOUSE = 0
 CAT = 1
 MOUSE_WIN = 1
 CAT_WIN = 2
 DRAW = 0
 n = len(graph)
 results = {}
 for i in range(1, n):
```



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Заключительный этап

Информатика 10 класс

Вариант 2

```
results[0, i, CAT] = results[0, i, MOUSE] = MOUSE_WIN
results[i, i, CAT] = results[i, i, MOUSE] = CAT_WIN
degree = {}
for mouse in range(1, n):
 for cat in range(1, n):
 degree[mouse, cat, MOUSE] = len(graph[mouse])
 degree[mouse, cat, CAT] = len(graph[cat]) - int(0 in graph[cat])
q = deque([state for state in results.keys()])
#BFS
while q:
 mouse, cat, turn = q.popleft()
 curResult = results[mouse, cat, turn]
 prevStates = []
 if turn == MOUSE:
 prevStates = [(mouse, prevCat, CAT) for prevCat in graph[cat]]
 else:
 prevStates = [(prevMouse, cat, MOUSE) for prevMouse in graph[mouse]]
 for prevState in prevStates:
 if prevState in results:
 continue
 prevMouse, prevCat, prevTurn = prevState
 if prevCat == 0:
 continue
 degree[prevState] -= 1
```



Многопрофильная олимпиада  
PTU МИРЭА  
2024/2025  
Заключительный этап  
Информатика 10 класс  
Вариант 2

```
isMoverWinner = ((curResult == MOUSE_WIN and prevTurn == MOUSE)
or
 (curResult == CAT_WIN and prevTurn == CAT))
if isMoverWinner or degree[prevState] == 0:
 results[prevState] = curResult
 q.append(prevState)
return results.get((1, 2, MOUSE), DRAW)
n = int(input()) # Количество вершин
graph = []
for _ in range(n):
 neighbors = list(map(int, input().split()))
 graph.append(neighbors)
result = cat_mouse_game(graph)
print(result)
...
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

### **Задача 1. Сокровища древнего храма**

**Ограничение по времени: 0.5 секунда**

**Ограничение по памяти: 256 МВ**

Вы — искатель приключений, исследующий древний храм. Внутри храма вы обнаружили несколько залов, каждый из которых заполнен сундуками с сокровищами. Каждый сундук содержит определённое количество золотых монет.

Ваша задача — собрать как можно больше монет, следуя древнему ритуалу:

В каждом зале вы должны выбрать сундук с наибольшим количеством монет и забрать его.

После этого из всех выбранных сундуков вы добавляете к своему счёту только те монеты, которые находятся в самом богатом сундуке.

Процесс повторяется до тех пор, пока все сундуки не будут опустошены.

Верните итоговое количество монет, которое вы смогли собрать.

#### **Входные данные:**

На вход подается:

Целое число  $m$  — количество залов.

Целое число  $n$  — количество сундуков в каждом зале.

Матрица размером  $m \times n$ , где каждая строка представляет зал, а каждый элемент строки — количество монет в соответствующем сундуке.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

**Формат ввода:**

...

$m$   $n$

<строка 1>

<строка 2>

...

<строка  $m$ >

...

**Выходные данные:**

Выведите одно целое число — итоговое количество монет, которое вы смогли собрать.

**Решение:**

```
```python
```

```
def calculate_treasure_score(nums):
```

```
    score = 0
```

```
    for i in range(len(nums)):
```

```
        nums[i].sort()
```

```
    for i in range(len(nums[0])):
```

```
        x = 0
```

```
        for j in range(len(nums)):
```

```
            x = max(x, nums[j][i])
```

```
    score += x
```

```
    return score
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

```
m, n = map(int, input().split())
nums = [list(map(int, input().split())) for _ in range(m)]
# Вычисление итогового счёта
result = calculate_treasure_score(nums)
print(result)
...
```

Задача 2. Каменные кучи

Ограничение по времени: 1.5 секунды

Ограничение по памяти: 128 МВ

Алёна и Борис играют в игру с кучами камней. Кучи выстроены в ряд, и каждая куча содержит положительное целое число камней. Игроки ходят по очереди, начиная с Алёны. На своём ходу игрок может взять камни из первых X оставшихся куч, где $1 \leq X \leq 2M$. После этого значение M обновляется: $M = \max(M, X)$. В начале игры $M = 1$. Игра заканчивается, когда все камни разобраны. Побеждает тот, кто соберёт больше камней.

Предполагая, что оба игрока играют оптимально, определите, сколько камней сможет собрать Алёна.

Формат ввода:

На первой строчке записано целое число n - количество камней в куче.

На второй строке подаётся массив целых чисел 'кучи', где каждый элемент представляет количество камней в соответствующей куче. Элементы разделены пробелом.

Длина массива: $1 \leq n \leq 100$

Количество камней в каждой куче: $1 \leq \text{кучи}[i] \leq 10^4$



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

Формат вывода:

Выведите одно целое число — максимальное количество камней, которое сможет собрать Алёна.

Решение:

```
```python
def max_stones(piles):
 n = len(piles)
 dp = [[0] * (n + 1) for _ in range(n)]
 suffix_sum = [0] * n
 suffix_sum[-1] = piles[-1]
 for i in range(n - 2, -1, -1):
 suffix_sum[i] = suffix_sum[i + 1] + piles[i]
 for i in range(n - 1, -1, -1):
 for m in range(1, n + 1):
 if i + 2 * m >= n:
 dp[i][m] = suffix_sum[i]
 else:
 for x in range(1, 2 * m + 1):
 dp[i][m] = max(dp[i][m], suffix_sum[i] - dp[i + x][max(m, x)])
 return dp[0][1]

n = int(input())
piles = list(map(int, input().split()))
print(max_stones(piles))
```
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

Задача 3. Художник

Ограничение по времени: 1 секунды

Ограничение по памяти: 64 МВ

Художник рисует на холсте размером $m \times n$ с помощью специальных правил:

На каждом шаге художник закрашивает сплошной прямоугольник одним цветом, который перекрывает все предыдущие цвета в этом прямоугольнике.

Каждый цвет можно использовать только один раз.

Вам дана сетка `targetGrid`, где `targetGrid[row][col]` — это цвет в позиции `(row, col)`.

Ваша задача — определить, может ли нарисовать такую сетку художник, соблюдая правила.

Формат ввода:

Первая строка содержит два целых числа m и n — размеры холста ($1 \leq m, n \leq 60$).

Следующие m строк содержат по n целых чисел, разделённых пробелами, — цвета в каждой ячейке холста ($1 \leq \text{targetGrid}[\text{row}][\text{col}] \leq 60$).

Формат вывода:

Выведите 'Да', если холст можно нарисовать с помощью волшебного художника, и 'Нет' в противном случае.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

Решение:

```
```python
def is_paintable(targetGrid):
 from collections import defaultdict
 from itertools import product
 def dp(color: int) -> bool:
 rows, cols = list(zip(*colors[color]))
 mn_r, mx_r, mn_c, mx_c = (min(rows), max(rows) + 1,
 min(cols), max(cols) + 1)
 for i, j in product(range(mn_r, mx_r), range(mn_c, mx_c)):
 if targetGrid[i][j] != color and targetGrid[i][j] > 0:
 return False
 for i in range(mn_r, mx_r):
 targetGrid[i][mn_c: mx_c] = [0]*(mx_c - mn_c)
 return True
 m, n = len(targetGrid), len(targetGrid[0])
 colors = defaultdict(list)
 for i, j in product(range(m), range(n)):
 colors[targetGrid[i][j]].append((i,j))
 color_set = set(colors)
 while color_set:
 tmp = set(filter(lambda x: dp(x), color_set))
 if not tmp: return 'Нет'
 color_set -= tmp
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

```
return 'Да'

m, n = map(int, input().split())
targetGrid = []
for _ in range(m):
 row = list(map(int, input().split()))
 targetGrid.append(row)
result = is_paintable(targetGrid)
print(result)
...
```

#### **Задача 4. Преобразование строки с помощью шаблона**

**Ограничение по времени: 1 секунды**

**Ограничение по памяти: 128 MB**

У вас есть строка `s`, состоящая из символов `'?'`, и целевая строка `target`. Изначально строка `s` имеет длину, равную длине строки `target`, и все её символы равны `'?'`.

Вы можете использовать шаблон (строку `stamp`), чтобы заменить символы `'?'` в строке `s` на соответствующие символы из шаблона. За один ход вы можете применить шаблон, начиная с любого индекса строки `s`, при условии, что шаблон полностью помещается в границы строки `s`.

Например, если шаблон равен `"abc"`, а целевая строка `target` равна `"ababc"`, то:

Применив шаблон начиная с индекса 0, получим `"abc???"`.

Применив шаблон начиная с индекса 2, получим `"???abc"`.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

Ваша задача — преобразовать строку `s` в целевую строку `target`, используя не более чем  $10 * \text{len}(\text{target})$  ходов. Если это невозможно, верните пустой массив.

#### Формат ввода:

Первая строка содержит строку `stamp` — шаблон ( $1 \leq \text{len}(\text{stamp}) \leq 1000$ ).

Вторая строка содержит строку `target` — целевая строка ( $1 \leq \text{len}(\text{target}) \leq 1000$ ).

#### Формат вывода

Выведите массив целых чисел, разделив каждый элемент массива пробелом, где каждое число — это индекс самого левого символа шаблона при каждом применении. Если преобразование невозможно, выведите пустую строку.

#### Решение:

```
```python
def moves_to_stamp(stamp, target):
    slen, tlen = len(stamp), len(target)
    res = []
    s_covers = set() # create permutation of cover on characters in stamp
    for i in range(slen):
        for j in range(slen - i):
            s_covers.add('#' * i + stamp[i:slen-j] + '#' * j)
    # print(s_covers)
    done = '#' * tlen
```



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 1

```
p = tlen - slen
while target != done:
    found = False
    for i in range(p, -1, -1):
        if target[i: i+slen] in s_covers:
            target = target[:i] + '#' * slen + target[i+slen:]
            res.append(i)
            found = True
    if not found:
        return []
    return res[::-1]
stamp = input().strip()
target = input().strip()
# Вычисление результата
result = moves_to_stamp(stamp, target)
# Вывод результата
print(' '.join(map(str, result)))
...
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

Задача 1. Путеводитель по древнему храму

Ты — исследователь древних храмов, который попал в загадочный храм. Этот храм имеет множество коридоров, которые представляют собой двумерную матрицу. В этой матрице каждый элемент представляет собой уровень опасности или ценности, который скрывается в том или ином месте храма.

Твоя цель — начать путешествие с любой ячейки в первом ряду и пройти как можно дальше по храму, двигаясь только в том случае, если впереди ты встречаешь место, которое представляет большую ценность, чем то, где ты находишься сейчас. Ты можешь двигаться на одну клетку вправо и вверх, вправо или вправо и вниз.

Но важно помнить: для того, чтобы продолжить путешествие, ты должен найти место, где ценность или уровень опасности выше, чем тот, где ты стоишь сейчас. Твоя задача — пройти по как можно большему числу таких мест.

Определи, сколько шагов ты можешь сделать, начиная с любой ячейки первого столбца храма.

Формат ввода:

Программа должна принимать матрицу `grid` из стандартного ввода в следующем формате:

Первая строка содержит два целых числа m и n — количество строк и столбцов матрицы, разделенных пробелом

Следующие m строк содержат по n целых чисел, разделенных пробелом — элементы матрицы.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

Формат вывода:

Программа должна вывести одно целое число — максимальное количество шагов, которое можно сделать, начиная с любой ячейки первого столбца.

Решение:

```
```python
```

```
def max_moves(grid: list[list[int]]) -> int:
```

```
 def dfs(i, j):
```

```
 nonlocal res
```

```
 val = grid[i][j]
```

```
 grid[i][j] = 0
```

```
 res = max(res, j)
```

```
 if res == n - 1:
```

```
 return
```

```
 for x in (i + 1, i - 1, i):
```

```
 if x >= 0 and x < m and j < n - 1 and grid[x][j + 1] > val:
```

```
 dfs(x, j + 1)
```

```
 res = 0
```

```
 m, n = len(grid), len(grid[0])
```

```
 for i in range(m):
```

```
 dfs(i, 0)
```

```
 return res
```



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

```
n, m = map(int, input().split())
grid = [list(map(int, input().split())) for _ in range(n)]
print(max_moves(grid))
...
```

## Задача 2. Игра с камнями

**Ограничение по времени: 2 секунды**

**Ограничение по памяти: 128 MB**

Алёна и Борис играют в игру с камнями, выстроенными в ряд. Каждый камень имеет свою ценность. Игроки ходят по очереди, начиная с Алёны. На своём ходу игрок может взять либо самый левый, либо самый правый камень из оставшихся. После этого игрок получает очки, равные сумме ценностей всех оставшихся камней. Игра заканчивается, когда все камни разобраны. Победителем считается тот, у кого больше очков.

Борис понял, что всегда проигрывает в этой игре, поэтому решил минимизировать разницу в очках. Алёна, напротив, хочет максимизировать эту разницу.

Ваша задача — определить разницу в очках между Алёной и Борисом, если оба играют оптимально.

### **Формат ввода:**

На первой строчке записано целое число  $n$  - количество камней в куче. На второй строчке подаётся массив целых чисел ``stones``, где каждый элемент представляет ценность соответствующего камня. Элементы разделены пробелом



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

Длина массива:  $2 \leq n \leq 100$

Ценность каждого камня:  $1 \leq \text{stones}[i] \leq 1000$

**Формат вывода:**

Выведите одно целое число — разницу в очках между Алёной и Борисом.

**Решение:**

```
```python
def stone_game_difference(stones):
    n = len(stones)
    flag = 1
    lis = list(stones)
    for i in range(1,n):
        for j in range(n - i):
            if flag:
                lis[j] = max(lis[j],lis[j+1])
            else:
                lis[j] = min(lis[j] + stones[j + i],lis[j+1] + stones[j])
        flag ^= 1
    return lis[0] if not flag else sum(stones) - lis[0]

input()
stones = list(map(int, input().split()))
print(stone_game_difference(stones))
```
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

### Задача 3. Встреча

**Ограничение по времени: 1 секунды**

**Ограничение по памяти: 128 МВ**

В Волшебном Лесу живут  $n$  магических существ, пронумерованных от 0 до  $n-1$ . Они решили организовать встречу у большого круглого камня, который может вместить любое количество существ. Однако каждое существо согласится прийти на встречу только в том случае, если сможет сидеть рядом со своим лучшим другом.

Каждое существо имеет своего лучшего друга, и это не может быть оно само. Задан массив `favorite`, где `favorite[i]` — это номер лучшего друга для существа с номером  $i$ .

Ваша задача — определить максимальное количество существ, которых можно пригласить на встречу, чтобы все они могли сидеть рядом со своими лучшими друзьями за круглым камнем.

#### Формат ввода:

Первая строка содержит целое число  $n$  — количество существ ( $2 \leq n \leq 10^5$ ).

Вторая строка содержит  $n$  целых чисел, разделённых пробелами, — массив `favorite`, где `favorite[i]` — это номер лучшего друга для существа  $i$  ( $0 \leq \text{favorite}[i] \leq n-1$ ,  $\text{favorite}[i] \neq i$ ).

#### Формат вывода:

Выведите одно целое число — максимальное количество существ, которых можно пригласить на встречу.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

**Решение:**

```
```python
def max_invited_creatures(n, favorite):
    from collections import defaultdict
    # Граф предпочтений
    graph = defaultdict(list)
    for i in range(n):
        graph[favorite[i]].append(i)
    # Поиск максимального цикла
    max_cycle = 0
    visited = [False] * n
    for i in range(n):
        if not visited[i]:
            cycle = []
            current = i
            while not visited[current]:
                visited[current] = True
                cycle.append(current)
                current = favorite[current]
            if current in cycle:
                cycle_length = len(cycle) - cycle.index(current)
                max_cycle = max(max_cycle, cycle_length)
    # Поиск пар взаимных предпочтений
    mutual_pairs = []
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Заключительный этап
Информатика 11 класс
Вариант 2

```
for i in range(n):
    if favorite[favorite[i]] == i and i < favorite[i]:
        mutual_pairs.append((i, favorite[i]))
# Поиск максимальной суммы цепочек для пар
max_chain_sum = 0
for pair in mutual_pairs:
    a, b = pair
    # Поиск максимальной длины цепочки для a
    chain_a = 0
    stack = [(a, 1)]
    while stack:
        node, depth = stack.pop()
        chain_a = max(chain_a, depth)
        for neighbor in graph[node]:
            if neighbor != b:
                stack.append((neighbor, depth + 1))
    # Поиск максимальной длины цепочки для b
    chain_b = 0
    stack = [(b, 1)]
    while stack:
        node, depth = stack.pop()
        chain_b = max(chain_b, depth)
        for neighbor in graph[node]:
            if neighbor != a:
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

```
stack.append((neighbor, depth + 1))

# Сумма цепочек
max_chain_sum += chain_a + chain_b

# Возвращаем максимум из длины цикла и суммы цепочек
return max(max_cycle, max_chain_sum)

n = int(input())
favorite = list(map(int, input().split()))
result = max_invited_creatures(n, favorite)
print(result)
...
```

Задача 4. Магические кристаллы

Ограничение по времени: 1 секунды

Ограничение по памяти: 128 MB

В далёкой волшебной стране магов есть древние магические кристаллы, которые хранят энергию. Каждый кристалл имеет определённый уровень энергии, представленный целым числом. Маги могут использовать заклинания, чтобы увеличить уровень энергии любого кристалла на 1. Однако каждое заклинание требует магической силы, и у магов есть ограниченный запас — не более k заклинаний.

Маги хотят исследовать последовательности кристаллов (подмассивы) и определить, сколько из них можно сделать "гармоничными". Последовательность кристаллов называется гармоничной, если уровень



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

энергии каждого следующего кристалла не меньше, чем у предыдущего. Например, последовательность $[2, 3, 3, 5]$ гармонична, а $[4, 2, 1]$ — нет.

Ваша задача — помочь магам определить, сколько подмассивов кристаллов можно сделать гармоничными, используя не более k заклинаний.

Формат ввода:

Первая строка содержит целое число n — количество кристаллов ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел, разделённых пробелами, — уровни энергии кристаллов ($1 \leq \text{nums}[i] \leq 10^9$).

Третья строка содержит целое число k — максимальное количество заклинаний ($1 \leq k \leq 10^9$).

Формат вывода:

Выведите одно целое число — количество подмассивов кристаллов, которые можно сделать гармоничными, используя не более k заклинаний.

Решение:

```
```python
def count_harmonic_subarrays(nums, k):
 from collections import deque
 nums = nums[::-1]
 res = 0
 q = deque()
 i = 0
 for j in range(len(nums)):
 while q and nums[q[-1]] < nums[j]:
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Заключительный этап

Информатика 11 класс

Вариант 2

```
 r = q.pop()
 l = q[-1] if q else i - 1
 k -= (r - l) * (nums[j] - nums[r])
 q.append(j)
 while k < 0:
 k += nums[q[0]] - nums[i]
 if q[0] == i:
 q.popleft()
 i += 1
 res += j - i + 1

 return res

Чтение входных данных
n = int(input())
nums = list(map(int, input().split()))
k = int(input())

Вычисление результата
result = count_harmonic_subarrays(nums, k)

Вывод результата
print(result)
...
```