



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

Задание 1. Чужая десятичная система

Представьте, что вы оказались на планете, где числа записываются не так, как на Земле. Там вместо привычных цифр от 4 до 9 используют буквы A, B, C, D, E, F . Эта система выглядит как смесь привычной десятичной и шестнадцатеричной систем, но её база — всё так же 10. Вам предстоит написать программу, которая сможет преобразовывать числа из привычной десятичной системы в эту нестандартную и обратно.

Формат ввода

Входные данные содержат два элемента:

1. Целое число m ($m = 1$ или $m = 2$):

- $m = 1$: конвертация из десятичной системы в нестандартную.
- $m = 2$: конвертация из нестандартной системы в десятичную.

2. В зависимости от значения m :

- Если $m = 1$, вводится одно целое неотрицательное число N ($0 \leq N \leq 10^9$).
- Если $m = 2$, вводится строка S , представляющая число в нестандартной системе счисления (длина строки не превышает 20 символов, символы только из множества $\{0, 1, 2, 3, A, B, C, D, E, F\}$).

Формат вывода

Выведите:

- Если $m = 1$, строку, представляющую число N в нестандартной системе счисления.
- Если $m = 2$, целое число, представляющее десятичное значение строки S .



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

Решение:

```
def decimal_to_custom(num):
    """Преобразует число из десятичной системы в нестандартную."""
    digits = "0123ABCDEF"
    result = []

    while num > 0:
        result.append(digits[num % 10])
        num //= 10

    return ''.join(reversed(result)) if result else '0'

def custom_to_decimal(s):
    """Преобразует число из нестандартной системы в десятичную."""
    digits = "0123ABCDEF"
    result = 0
    for char in s:
        result = result * 10 + digits.index(char)
    return result

if __name__ == '__main__':
    m = int(input())
    if m == 1:
        print(decimal_to_custom(int(input())))
    else:
        print(custom_to_decimal(input()))
```

Задание 2. Реверсивный маятник

У вас есть массив из n целых чисел. Необходимо преобразовать его так, чтобы:

1. Наименьший элемент оказался в **центре** массива.
2. Следующий по величине элемент оказался **справа от центра**, затем следующий — **слева от центра**, и так далее.
3. Таким образом, элементы должны чередоваться между левым и правым направлениями относительно центра массива.

Если n чётное, наименьший элемент окажется на позиции $n/2$, а правый элемент начнётся с $n/2+1$.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

Формат ввода:

В первой строке входного файла содержится число n ($1 \leq n \leq 30000$).

Во второй строке через пробел записаны n целых чисел, по абсолютной величине не превышающих 30000.

Формат вывода:

Выведите элементы преобразованного массива в одну строку, разделённые пробелами.

Решение:

```
if __name__ == '__main__':  
    n = int(input())  
    array = list(map(int, input().split()))  
    array.sort()  
    result = [0] * n  
    left = (n - 1) // 2  
    right = left + 1  
    toggle = True  
  
    for i in range(n):  
        if toggle:  
            result[left] = array[i]  
            left -= 1  
        else:  
            result[right] = array[i]  
            right += 1  
        toggle = not toggle  
  
    print(' '.join(map(str, result)))
```

Задание 3. Пирамиды

На плоскости нужно нарисовать K пирамид разной высоты, расположенных в ряд. Все пирамиды стоят на одной горизонтальной линии. Каждая пирамида представляет собой равнобедренный треугольник, вершина которого находится в центре строки, а основание занимает строку на нижней линии. Расстояние между соседними пирамидами — один пробел.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

Формат ввода

1. Первое число K ($1 \leq K \leq 50$) — количество пирамид.
2. Через пробел указаны K чисел $H_1, H_2, \dots, H_K$ ($1 \leq H_i \leq 50$) — высоты пирамид

Формат вывода

Выведите прямоугольник, состоящий из пробелов и * (звёздочек), который изображает K пирамид, расположенных в ряд.

- Все пирамиды выровнены по нижнему краю.
- Ширина основания каждой пирамиды равна $2 \times H_i - 1$.
- Между каждой парой соседних пирамид — ровно один пробел.

Решение:

```
def pyramids_in_row(k, heights):  
    # Определяем максимальную высоту среди всех пирамид  
    max_height = max(heights)  
  
    # Генерируем строки для итогового рисунка  
    rows = []  
    for level in range(max_height - 1, -1, -1): # Стартуем с верхнего уровня  
        row = []  
        for h in heights:  
            if level < h: # Если текущий уровень входит в пирамиду  
                stars = 1 + 2 * (h - level - 1) # Количество звёздочек для уровня  
                row.append('*' * stars).center(2 * h - 1, ' ')  
  
            else: # Уровень выше вершины пирамиды  
                row.append(' ' * (2 * h - 1))  
        # Соединяем пирамиды с одной точкой между ними  
        rows.append(' '.join(row))  
  
    return rows  
  
def main():
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

```
k = int(input())
heights = list(map(int, input().split()))
result = pyramids_in_row(k, heights)
for line in result:
    print(line)

main()
```

Задание 4. Последовательность

В древнем храме Логике хранится артефакт — **Кодекс Равновесия**, поддерживающий баланс между мирами. Этот артефакт работает на магических символах — скобках: {}, [], (). Однако недавно коды начали путаться, и баланс оказался под угрозой. Если ошибиться и ввести некорректную последовательность, реальности пересекутся, и миры погрузятся в хаос.

Вы — ученик храма, избранный, чтобы восстановить порядок! Ваш долг — проверить последовательности магических символов и отобрать только те, которые корректны. Только правильные последовательности активируют Кодекс Равновесия.

Последовательность скобок называется корректной, если выполняются следующие условия:

1. **Открывающие скобки должны быть закрыты теми же типами скобок.** То есть, если открывается скобка типа {, она должна быть закрыта точно такой же скобкой}.
2. **Открывающие скобки должны быть закрыты в правильном порядке.** Например, если открывается скобка (, то сначала должна быть закрыта она же, а затем другие скобки.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс

3. Каждая закрывающая скобка должна иметь соответствующую открывающую скобку того же типа. То есть, если встречается закрывающая скобка, она должна иметь пару среди открывающих скобок.

Формат ввода

Строка S (длиной от 1 до 10^5), состоящая только из символов: (,), , , [,].

Формат вывода

Выведите строку "YES", если последовательность корректна, и "NO", если нет.

Решение:

```
def is_balanced(s: str) -> str:
    stack = []
    brackets = {'(': ')', '{': '}', '[': ']'}

    for char in s:
        if char in brackets.values(): # если это открывающая скобка
            stack.append(char)
        elif char in brackets: # если это закрывающая скобка
            if not stack or stack.pop() != brackets[char]:
                return "NO"

    return "YES" if not stack else "NO"

# Чтение входной строки
s = input().strip()
print(is_balanced(s))
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 1

Задание 1. Перевод чисел между системами счисления

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 МВ

Дано число N в десятичной системе счисления. Переведите его в систему счисления с основанием B (где $2 \leq B \leq 36$).

Входные данные:

На вход подаются два целых числа: N ($0 \leq N \leq 10^9$) и B ($2 \leq B \leq 36$)

Выходные данные:

Выведите одно число — сумму всех цифр двоичного представления числа N .

Ввод	Вывод
255 16	FF
100 2	1100100
987654 36	I62U

Решение:

```
def decimal_to_base(N, B):  
    if N == 0:  
        return '0'  
  
    digits = []  
  
    while N:  
        remainder = N % B  
        if remainder < 10:  
            digits.append(str(remainder))  
        else:  
            digits.append(chr(ord('A') + remainder - 10))  
        N = N // B  
  
    return ''.join(reversed(digits))
```



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Отборочный этап

Информатика 9 класс

Вариант 1

```
N //= B

return ''.join(reversed(digits))

N, B = map(int, input().split())

result = decimal_to_base(N, B)
print(result)
```

Задание 2. Странная последовательность

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 МВ

Дана последовательность чисел $(a_1, a_2, \dots, a_n)$. Вам нужно найти минимальное значение k , такое что сумма квадратов разностей $(a_i - k)^2$ для всех i минимальна.

Входные данные:

В первой строке дано одно число n ($1 \leq n \leq 10^5$). Во второй строке содержится последовательность из n целых чисел $a_1, a_2, \dots, a_n$, где каждое число по модулю не превышает 10^9 .

Выходные данные:

Выведите минимальное значение k .

Ввод	Вывод
5 1 2 3 4 5	3
6 -10 -5 0 5 10 15	2.5
3 1 1 1	1



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 1

Решение:

```
def find_k_min(arr):  
    arr.sort()  
    n = len(arr)  
    if n % 2 == 1:  
        k = arr[n // 2]  
    else:  
        k = (arr[n // 2 - 1] + arr[n // 2]) / 2  
    return float(k)  
  
n = int(input())  
sequence = list(map(int, input().split()))  
print(find_k_min(sequence))
```

Задание 3. Таинственный артефакт

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

В древнем храме археологи нашли свиток с древними надписями, записанными в виде строки. Важно выяснить, уникальна ли каждая буква в надписи. Определите, есть ли в строке повторяющиеся символы.

Входные данные:

Одна строка s , содержащая от 1 до 10^5 строчных латинских букв – это текст на свитке.

Выходные данные:

Выведите "YES", если в строке есть повторяющиеся символы, иначе выведите "NO".



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Отборочный этап

Информатика 9 класс

Вариант 1

Ввод	Вывод
sacred	NO
ancient	YES
hieroglyph	YES

Решение:

```
def has_duplicate_characters(s):  
    seen = set()  
    for char in s:  
        if char in seen:  
            return "YES"  
        seen.add(char)  
    return "NO"  
  
print(has_duplicate_characters(input()))
```

Задание 4. Остроконечные амулеты

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

В древней библиотеке археологи нашли три загадочных амулета, каждый из которых имеет форму треугольника. В легенде говорится, что амулеты обладают магической силой, если они образуют остроугольный треугольник. Помогите археологам выяснить, обладают ли амулеты магической силой.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 1

Входные данные:

Вам даны три точки на плоскости: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Во второй строке даны три целых числа (x_2, y_2) – координаты точки B . В третьей строке даны три целых числа (x_3, y_3) – координаты точки C .

Выходные данные:

Выведите "YES", если амулеты образуют остроугольный треугольник, иначе выведите "NO". Если точки совпадают, также выведите "NO".

Ввод	Вывод
0 0 1 0 0 1	YES
0 0 1 1 2 2	NO

Решение:

```
def is_acute_triangle(x1, y1, x2, y2, x3, y3):  
    def length_squared(x1, y1, x2, y2):  
        return (x2 - x1) ** 2 + (y2 - y1) ** 2  
  
    # Проверяем на совпадение точек  
    if (x1 == x2 and y1 == y2) or (x1 == x3 and y1 == y3) or (x2 == x3 and  
y2 == y3):  
        return "NO"  
  
    # Проверяем на коллинеарность  
    D = (x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1)  
    if D == 0:  
        return "NO"
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 1

```
# Вычисляем квадраты длин сторон
ab2 = length_squared(x1, y1, x2, y2)
bc2 = length_squared(x2, y2, x3, y3)
ca2 = length_squared(x3, y3, x1, y1)

# Проверяем, все ли углы острые
if (ab2 + bc2 > ca2) and (ab2 + ca2 > bc2) and (bc2 + ca2 > ab2):

    return "YES"
else:
    return "NO"

if __name__ == '__main__':

    x1, y1 = map(int, input().split())
    x2, y2 = map(int, input().split())
    x3, y3 = map(int, input().split())

    print(is_acute_triangle(x1, y1, x2, y2, x3, y3))
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 2

Задание 1. Определение валидности числа в системе счисления

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

Дано число X в системе счисления с основанием B . Определите, является ли это число корректным в данной системе счисления.

Входные данные:

На вход подаются строка X и целое число B ($2 \leq B \leq 36$)

Выходные данные:

Если число X является корректным в системе счисления с основанием B , выведите "YES", иначе "NO".

Ввод	Вывод
101010 2	YES
Z9 16	NO

Решение:

```
def is_valid_number_in_base(X, B):  
    try:  
        int(X, B)  
        return "YES"  
    except ValueError:  
        return "NO"  
  
if __name__ == '__main__':  
    data = input().split()  
    X = data[0]  
    B = int(data[1])  
    print(is_valid_number_in_base(X, B))
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 9 класс

Вариант 2

Задание 2. Секретное число пиратов

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

Пираты на таинственном острове нашли старинную карту с числами, которые нужно расшифровать. На карте указано несколько чисел, каждое из которых является произведением двух простых чисел. Вам нужно определить, сколько различных простых чисел участвует в этих произведениях.

Входные данные:

В первой строке дано целое число n ($1 \leq n \leq 1000$) – количество чисел на карте. Во второй строке даны n целых чисел, каждое из которых представляет произведение двух простых чисел. Числа не превышают 10^6 .

Выходные данные:

Выведите одно целое число – количество различных простых чисел, которые встречаются в этих произведениях.

Ввод	Вывод
3 6 10 35	4
4 12 18 30 45	3
6 15 21	3

Решение:

```
def prime_factors(n):  
    factors = []  
    while n % 2 == 0:  
        factors.append(2)  
        n //= 2
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 2

```
for i in range(3, int(n ** 0.5) + 1, 2):
    while n % i == 0:
        factors.append(i)
        n //= i

# Если n остаётся больше 2, то оно простое
if n > 2:
    factors.append(n)
return factors

def answer(n, data):
    answ = set()
    for el in data:
        answ.update(prime_factors(el))
    return len(answ)

if __name__ == '__main__':
    n = int(input())
    data = map(int, input().split())
    print(answer(n, data))
```

Задание 3. Испорченный телепорт

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

Во время телепортации через космический портал происходит сбой, и буквы в заклинании перемешиваются. Однако в портале осталась возможность поменять все вхождения одного символа на другой. Помогите волшебникам восстановить порядок, заменив один символ на другой.

Входные данные:

В первой строке дана строка s от 1 до 10^6 , содержащая строчные латинские буквы – заклинание. Во второй строке указаны два символа c_1 и c_2 – строчные латинские буквы, которые нужно заменить.

Выходные данные:

Выведите строку s , где все вхождения символа c_1 заменены на c_2 .



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Отборочный этап

Информатика 9 класс

Вариант 2

Ввод	Вывод
abracadabra a o	obrocodobro
Wizardry i z	wzzardry
dragon d b	bragon

Решение:

```
def replace_character(spell: str, c1: str, c2: str) -> str:
    return spell.replace(c1, c2)

if __name__ == '__main__':
    s = input().strip() # Считываем строку заклинания
    c1, c2 = input().strip().split() # Считываем два символа для замены
    result = replace_character(s, c1, c2)
    print(result)
```

Задание 4. Магическая сортировка

Ограничение по времени: 1 секунда

Ограничение по памяти: 128 МВ

Маги из волшебного королевства нашли древние свитки с изображениями треугольников. Они хотят знать, какие из треугольников имеют большую магическую силу, что определяется их площадью. Помогите магам отсортировать треугольники по убыванию их магической силы, чтобы они могли использовать самые могущественные.

Условие:

Вам даны n треугольников на плоскости, каждый из которых задаётся тремя тройками координат вершин. Определите порядок треугольников по



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 9 класс

Вариант 2

убыванию их площади. Если несколько треугольников имеют одинаковую площадь, их порядок определить в порядке возрастания номера треугольника.

Входные данные:

Первая строка содержит целое число n ($1 \leq n \leq 1000$) – количество треугольников. Далее следуют строки с координатами трёх вершин каждого треугольника. Каждая строка содержит шесть целых чисел, представляющих координаты трёх вершин треугольника.

Выходные данные:

Выведите n строк, каждая из которых содержит номера треугольников в порядке убывания их площади. Нумерация начинается с 1.

Ввод	Вывод
3	2
0 0 1 0 0 1	1
0 0 2 0 2 2	3
1 0 0 0 0 1	
2	1
0 0 2 0 2 2	2
0 0 0 1 1 1	



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 9 класс
Вариант 2

Решение:

```
def triangle_area(x1, y1, x2, y2, x3, y3):  
    return abs(x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / 2  
  
def main(n, data):  
    triangles = []  
  
    for i in range(n):  
        coords = list(map(int, data[i].split()))  
        x1, y1, x2, y2, x3, y3 = coords  
        area = triangle_area(x1, y1, x2, y2, x3, y3)  
        triangles.append((area, i + 1)) # Сохраняем площадь и номер треугольника  
  
    # Сортируем по убыванию площади, а при равенстве по возрастанию индекса  
    triangles.sort(key=lambda x: (-x[0], x[1]))  
  
    # Выводим номера треугольников в нужном порядке  
  
    answ = []  
    for _, index in triangles:  
        answ.append(index)  
  
    return '\n'.join(map(str, answ))  
  
if __name__ == "__main__":  
    n = int(input())  
    data = [input() for _ in range(n)]  
    print(main(n, data))
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Задание 1. Ключ к равновесию

Давным-давно, в далёкой галактике, существовали два мира — Мир Альфа и Мир Бета. Эти два мира были тесно связаны, но их культуры, технологии и даже математика сильно различались. Математическая система каждого мира базировалась на своих числовых основаниях. Мир Альфа и Мир Бета долгое время жили в гармонии, но однажды баланс между ними нарушился. Чтобы восстановить равновесие, необходимо найти уникальный **ключ баланса**, обозначенный как x . Этот ключ является числовой точкой пересечения двух миров и может быть выражен в специальной системе счисления, принятой для меж мировых договоров.

Учёные обоих миров провели вычисления и описали структуру мира в виде уравнений. Ваша задача — найти ключ x , который удовлетворяет обоим уравнениям, и представить его в общей системе счисления.

$$a_1x + b_1 = c_1, a_2x + b_2 = c_2$$

Формат ввода:

На вход подаётся три строки:

1. Первая строка: $a_1 \ b_1 \ c_1 \ B_1$ — коэффициенты уравнения из Мира Альфа, записанные в системе счисления B_1 .

2. Вторая строка: $a_2 \ b_2 \ c_2 \ B_2$ — коэффициенты уравнения из Мира Бета, записанные в системе счисления B_2 .

3. Третья строка: B_x — система счисления, в которой нужно вывести ключ x .

Все числа задаются строками, корректными для указанных систем счисления.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Формат вывода:

Выведите одно число x , являющееся целым значением, записанное в системе счисления B_x .

Решение:

```
def base_to_decimal(num_str, base):
    """
    Преобразует строку num_str из системы счисления base в десятичную.
    """
    return int(num_str, base)

def decimal_to_base(num, base):
    """
    Преобразует десятичное число num в строку в системе счисления base.
    """
    if num == 0:
        return "0"
    digits = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    result = ""
    while num > 0:
        result = digits[num % base] + result
        num //= base
    return result

def solve():
    import sys

    input = sys.stdin.read
    data = input().splitlines()

    # Читаем первую строку (Мир Альфа)
    a1, b1, c1, B1 = data[0].split()
    a1, b1, c1 = base_to_decimal(a1, int(B1)), base_to_decimal(b1, int(B1)),
    base_to_decimal(c1, int(B1))

    # Читаем вторую строку (Мир Бета)
    a2, b2, c2, B2 = data[1].split()
    a2, b2, c2 = base_to_decimal(a2, int(B2)), base_to_decimal(b2, int(B2)),
    base_to_decimal(c2, int(B2))

    # Читаем третью строку (Требуемая система счисления Bx)
    Bx = int(data[2])
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

```
x1 = (c1 - b1) / a1
x2 = (c2 - b2) / a2

if x1 != x2 or not x1.is_integer():
    print("Нет решения")
    return

x = int(x1)

# Выводим x в системе счисления Bx
print(decimal_to_base(x, Bx))

if __name__ == "__main__":
    solve()
```

Задание 2. Казна

Королевский казначей поручил вам ответственное задание. Перед вами два сундука, в каждом из которых хранятся монеты, обозначающие цифры числа. Порядок размещения монет в сундуках имеет свою специфику: монеты кладутся в сундук начиная с единиц, затем десятков, сотен и так далее. Это означает, что:

- Первая монета на дне сундука соответствует единицам числа.
- Вторая монета — десяткам,
- Третья — сотням, и так далее.

При извлечении монет из сундука их порядок становится обратным: мы видим цифры начиная с самой старшей (сотен, тысяч) и заканчивая самой младшей (единицами).

Ваша задача:

Сформировать новый сундук с монетами, в котором будет храниться сумма двух сундуков.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Формат ввода:

На вход подаются следующие данные:

1. Количество монет в первом сундуке.
2. Количество монет во втором сундуке.
3. Содержимое первого сундука (через пробел) — записано в том порядке, в котором монеты были положены в сундук.
4. Содержимое второго сундука (через пробел) — записано аналогичным образом.

Формат вывода:

Выведите одну строку, содержащую цифры нового сундука (через пробел) в порядке, соответствующем их добавлению: сначала единицы, затем десятки, сотни и так далее.

Решение:

```
def sum_chests(n1, n2, chest1, chest2):  
    """  
    Складывает два числа, представленные как перевёрнутые списки цифр.  
  
    Числа хранятся в сундуках в обратном порядке (единицы первыми).  
    Функция складывает их и возвращает результат также в перевёрнутом порядке.  
  
    :param n1: int — количество монет в первом сундуке.  
    :param n2: int — количество монет во втором сундуке.  
    :param chest1: list[int] — содержимое первого сундука (в порядке помещения).  
    :param chest2: list[int] — содержимое второго сундука (в порядке помещения).  
    :return: list[int] — результат сложения в формате, аналогичном сундукам.  
    """  
  
    num1 = int(''.join(map(str, reversed(chest1))))  
    num2 = int(''.join(map(str, reversed(chest2))))  
    total = num1 + num2  
    return ' '.join([el for el in str(total)[::-1]])
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

```
def main():
    """Считывает входные данные, вычисляет сумму и выводит результат."""
    n1 = int(input().strip()) # Количество цифр в первом сундуке
    n2 = int(input().strip()) # Количество цифр во втором сундуке
    chest1 = list(map(int, input().split())) # Цифры первого сундука
    chest2 = list(map(int, input().split())) # Цифры второго сундука

    result = sum_chests(n1, n2, chest1, chest2) # Вычисляем сумму
    print(result) # Выводим результат

if __name__ == "__main__":
    main()
```

Задание 3. Римские

Семь различных символов представляют римские цифры со следующими значениями:

Символ	Значение
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Римские цифры формируются путем записи значений десятичных разрядов в порядке от большего к меньшему. При этом выполняются следующие правила преобразования:



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

1. Если значение разряда не начинается с 4 или 9, выбирается символ с максимальным значением, которое можно вычесть из числа, этот символ добавляется в результат, его значение вычитается, а оставшаяся часть числа снова преобразуется в римские цифры.

2. Если значение разряда начинается с 4 или 9, используется вычитательная форма записи: например, 4 — это на 1 (I) меньше 5 (V), то есть IV, а 9 — на 1 (I) меньше 10 (X), то есть IX. Допустимые вычитательные формы: IV (4), IX (9), XL (40), XC (90), CD (400), CM (900).

3. Символы, представляющие степени десяти (I, X, C, M), могут повторяться подряд не более 3 раз. Символы V, L и D нельзя повторять. Если требуется записать символ 4 раза, используется вычитательная форма.

Требуется: Написать программу, которая по заданному числу n ($1 \leq n \leq 3999$) выводит его запись римскими цифрами.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Решение:

```
def int_to_roman(num):  
    """  
    Преобразует целое число в римскую запись.  
  
    Использует жадный алгоритм, начиная с наибольшего римского значения и уменьшая число.  
  
    :param num: int – число в десятичной системе (1 ≤ num ≤ 3999).  
    :return: str – римская запись числа.  
    """  
    roman_map = [  
        (1000, "M"), (900, "CM"), (500, "D"), (400, "CD"),  
        (100, "C"), (90, "XC"), (50, "L"), (40, "XL"),  
        (10, "X"), (9, "IX"), (5, "V"), (4, "IV"),  
        (1, "I")  
    ]  
    result = ""  
  
    for value, symbol in roman_map:  
        while num >= value:  
            result += symbol  
            num -= value  
  
    return result  
  
def main():  
    """Считывает число с ввода, преобразует его в римскую запись и выводит результат."""  
    num = int(input().strip()) # Читаем число  
    print(int_to_roman(num)) # Преобразуем и выводим  
  
if __name__ == "__main__":  
    main()
```

Задание 4. Поворот матрицы на 90 градусов

Представьте, что у вас есть квадратное изображение, представленное в виде двумерной матрицы размером $n \times n$. Ваша задача — повернуть это изображение на 90 градусов по часовой стрелке.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Условия:

- Поворот необходимо выполнить на месте, то есть изменить входную матрицу напрямую.
- Запрещено создавать дополнительную двумерную матрицу для выполнения поворота.

Пример:

Входная матрица:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Результаты после поворота:

$$\begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

Формат ввода:

1. **Число n** — размер квадратной матрицы ($1 \leq n \leq 1000$).
2. Далее **n строк**, каждая из которых содержит n целых чисел, разделённых пробелом — элементы матрицы.

Формат вывода:

После выполнения поворота на 90 градусов по часовой стрелке, необходимо вывести n строк: Каждая строка должна содержать n чисел, разделённых пробелом — элементы повернутой матрицы.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс

Решение:

```
def rotate_matrix(matrix):  
    """  
    Поворачивает квадратную матрицу на 90 градусов по часовой стрелке.  
  
    Аргументы:  
    matrix -- квадратная матрица (список списков), которую нужно повернуть  
  
    Возвращает:  
    matrix -- повернутая матрица  
    """  
    n = len(matrix)  
  
    # Транспонируем матрицу (меняем строки на столбцы)  
    for i in range(n):  
        for j in range(i, n):  
            matrix[i][j], matrix[j][i] = matrix[j][i], matrix[i][j]  
  
    # Реверсируем каждую строку, чтобы завершить поворот на 90 градусов  
    for row in matrix:  
        row.reverse()  
  
    return matrix  
  
if __name__ == '__main__':  
    """  
    Вводит размер матрицы и саму матрицу, после чего вызывает функцию  
    для поворота матрицы на 90 градусов и выводит результат.  
    """  
    n = int(input())  
    matrix = [input().split() for _ in range(n)]  
  
    # Поворачиваем матрицу  
    rotated_matrix = rotate_matrix(matrix)  
  
    for row in rotated_matrix:  
        print(" ".join(row))
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 10 класс

Вариант 1

Задание 1. Лабиринт с сокровищами

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

Дима увлекается криптографией и программированием. Он придумал новый способ шифрования изображений — для этого он переворачивает изображение по вертикальной оси и замаскировывает его пиксели, чтобы сделать его неузнаваемым. Однако его программа иногда дает сбой и не всегда правильно выполняет операцию переворота.

Вам необходимо помочь Диме выявить ошибки. Для этого вам даны исходное изображение и зашифрованное. Ваша задача — проверить, правильно ли программа Димы перевернула строки изображения, и, если нет, подсчитать, сколько строк было зашифровано неверно.

Входные данные:

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 100$) — высоту и ширину изображения.

Следующие n строк содержат по m целых чисел от 0 до 255 — это исходное изображение, где каждое число — яркость соответствующего пикселя. Далее идет пустая строка, а после неё — n строк с зашифрованной версией изображения, где каждая строка также состоит из m чисел.

Выходные данные:

Выведите одно целое число — количество строк, которые были неправильно перевернуты при шифровании.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 1

Ввод	Вывод
4 5 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 50 40 30 20 10 100 90 80 70 60 140 130 120 110 150 200 190 180 170 160	2

Решение:

```
def count_incorrect_rows(n, m, original, encrypted):
    incorrect_count = 0
    for i in range(n):
        if original[i] != encrypted[i][::-1]: # Сравниваем оригинальную строку с
перевёрнутой зашифрованной
            incorrect_count += 1
    return incorrect_count

n, m = map(int, input().split())
original = [list(map(int, input().split())) for _ in range(n)]
input() # Пропускаем пустую строку

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

# Подсчет некорректных строк
result = count_incorrect_rows(n, m, original, encrypted)
print(result)
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 1

Задание 2. Сокровища двух пиратов

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 МВ

Два пирата нашли карты, ведущие к сокровищам на необитаемом острове. Каждый пират составил свой список уникальных сокровищ, которые они планируют найти. Однако, чтобы не драться из-за одних и тех же сокровищ, им нужно выяснить, какие уникальные сокровища находятся в обоих списках. Помогите пиратам определить, сколько сокровищ они могут поделить без конфликтов.

Входные данные:

В первой строке содержится целое число n ($1 \leq n \leq 1000$) — количество сокровищ в первом списке, затем n целых чисел.

Во третьей строке содержится целое число m ($1 \leq m \leq 1000$) — количество сокровищ во втором списке, затем m целых чисел.

Выходные данные:

Выведите одно целое число — количество уникальных сокровищ, которые есть в обоих списках.

Ввод	Вывод
5 1 2 3 4 5 4 3 4 5 6	2
3 1 1 1 2 2 2	1



Многопрофильная олимпиада

PTU МИРЭА

2024/2025

Отборочный этап

Информатика 10 класс

Вариант 1

Решение:

```
n = int(input())
data1 = set(map(int, input().split()))
m = int(input())
data2 = set(map(int, input().split()))
print(len(data1.symmetric_difference(data2)))
```

Задание 3. Сбор урожая

Ограничение по времени: 0.5 секунд

Ограничение по памяти: 16 MB

На ферме у фермера Пети растут три вида овощей: картошка весит X грамм, морковка — Y грамм, а капуста — Z грамм. Петя хочет собрать урожай в мешок, вес которого точно равен W грамм. Помогите ему определить, сколько различных способов он может собрать урожай.

Входные данные:

В единственной строке содержится четыре целых числа X , Y , Z и W ($1 \leq X, Y, Z \leq 100$, $1 \leq W \leq 1000$).

Выходные данные:

Выведите одно целое число — количество способов собрать урожай весом ровно W грамм.

Ввод	Вывод
10 25 15 40	3



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 1

Решение:

```
def count_ways(X, Y, Z, W):  
    # Массив dp для хранения количества способов собрать веса от 0 до W  
    dp = [0] * (W + 1)  
    dp[0] = 1 # Один способ собрать вес 0 (ничего не брать)  
    # В массиве dp обновляем для каждого овоща  
    for weight in (X, Y, Z): # Перебираем веса овощей  
        for i in range(weight, W + 1):  
            dp[i] += dp[i - weight]  
    return dp[W]  
  
X, Y, Z, W = map(int, input().strip().split())  
print(count_ways(X, Y, Z, W))
```

Задание 4. Учет фонарей в лесу

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

В лесу «Зелёный Оазис» расположено N полян и M тропинок, которые соединяют некоторые из полян. Лесной совет решил установить фонари на каждой тропинке перед входом на каждую поляну, чтобы обеспечить безопасность ночных прогулок. Напишите программу, которая посчитает, сколько фонарей должно быть установлено на каждой из полян. Поляны пронумерованы от 1 до N .

Входные данные:

На первой строчке записано два числа N и M ($0 < N \leq 100$, $0 \leq M \leq N \cdot (N - 1)/2$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что поляна i соединена с поляной j тропинкой. Можно считать, что любые две поляны соединены не более чем одной тропинкой. Нет тропинок от поляны i к самой себе.



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 10 класс

Вариант 1

Выходные данные:

Выведите N чисел: k -ое число означает количество фонарей на k -ой поляне.

Ввод	Вывод
7 10	3 3 2 2 5 2 3
5 1	
3 2	
7 1	
5 2	
7 4	
6 5	
6 4	
7 5	
2 1	
5 3	



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 1

Решение:

```
def count_traffic_lights(N, M, tunnels):
    # Инициализируем список для хранения количества фонарей на каждой поляне
    traffic_lights = [0] * (N + 1) # Используем индексирование с 1, поэтому создаем
    массив размером N+1

    # Обрабатываем каждую тропинку
    for i, j in tunnels:
        traffic_lights[i] += 1 # Увеличиваем счетчик фонарей для поляны i
        traffic_lights[j] += 1 # Увеличиваем счетчик фонарей для поляны j

    return traffic_lights[1:] # Возвращаем список без первого элемента (индекс 0 не
    используется)

def main():
    # Считываем количество полей и тропинок
    N, M = map(int, input().strip().split())
    # Считываем список тропинок
    tunnels = [tuple(map(int, input().strip().split())) for _ in range(M)]
    # Подсчитываем количество фонарей на каждой поляне
    result = count_traffic_lights(N, M, tunnels)
    # Выводим результат в виде последовательности чисел, разделенных пробелами
    print(' '.join(map(str, result)))

if __name__ == '__main__':
    main()
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 10 класс

Вариант 2

Задание 1. Зеркальное отражение изображения

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

Миша решил изучить работу QR-кодов и написал программу, которая сканирует изображение QR-кода размером $n \times n$. Однако при сканировании некоторые строки могут оказаться зашумленными. Ваша задача — помочь Мише проверить, является ли QR-код симметричным относительно его главной диагонали. Если код несимметричен, программа должна выявить количество строк, где зашифрованные пиксели не совпадают с симметричными элементами относительно главной диагонали.

Входные данные:

Первая строка содержит одно целое число n ($1 \leq n \leq 100$) — размер QR - кода (матрицы). Следующие n строк содержат по n символов: либо “0” (черный пиксель), либо “1” (белый пиксель), представляя изображение QR - кода.

Выходные данные:

Выведите одно целое число — количество строк, где зашифрованные пиксели не совпадают со своими симметричными элементами относительно главной диагонали.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

Ввод	Вывод
4	1
1 0 1 0	
0 1 0 1	
1 0 1 1	
0 1 0 1	

Примечание:



Решение:

```
def count_non_symmetric_rows(n, qr_code):  
    """  
    Функция проверяет симметричность QR-кода относительно главной диагонали  
    и подсчитывает количество строк, где есть расхождения.  
  
    :param n: размер QR-кода (матрицы)  
    :param qr_code: список строк, представляющий QR-код
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

```
:return: количество строк с несовпадающими элементами
"""
non_symmetric_rows = 0 # Счетчик строк с несовпадениями

for i in range(n):
    for j in range(i + 1, n): # Проверяем только верхний треугольник матрицы
        if qr_code[i][j] != qr_code[j][i]:
            non_symmetric_rows += 1 # Если найдено несовпадение, увеличиваем счетчик
            break # Достаточно одного несовпадения в строке, чтобы её учесть

return non_symmetric_rows

n = int(input()) # Считываем размер QR-кода
qr_code = [input().strip() for _ in range(n)] # Считываем сам QR-код

print(count_non_symmetric_rows(n, qr_code))
```

Задание 2. Уникальные слова в заклинании

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 MB

В мире магии древние заклинания хранятся в древних свитках. Один из магов нашёл свиток с заклинанием, которое он собирается использовать для своей следующей битвы. Однако перед тем, как произнести заклинание, он хочет убедиться, что использует все уникальные слова из свитка, чтобы увеличить его мощь. Помогите магу определить, сколько уникальных слов содержится в заклинании.

Входные данные:

Единственная строка, содержащая слова заклинания (длина строки не превышает 1000 символов).

Выходные данные:

Выведите одно целое число — количество уникальных слов в заклинании.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

Ввод	Вывод
Это пример строки с примерами	5
Слова в строке: Слово слово СЛОВО	4
Привет мир привет	2

Решение:

```
import sys

def count_unique_words(spell):
    """
    Функция подсчитывает количество уникальных слов в заклинании.

    :param spell: строка, содержащая заклинание
    :return: количество уникальных слов
    """
    words = spell.lower().split() # Приводим строку к нижнему регистру и разбиваем на
    слова
    return len(set(words)) # Используем множество для поиска уникальных слов

spell = sys.stdin.read().strip()
print(count_unique_words(spell))
```

Задание 3. Формирование кодов доступа

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 МВ

В системе безопасности у нас есть код доступа, состоящий из цифр. Код доступа имеет длину N , и требуется узнать, сколько различных трехзначных кодов можно сформировать, удаляя некоторые цифры из данного кода. Помогите пользователю определить количество уникальных трехзначных кодов.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

Входные данные:

Содержит строку, представляющую натуральное число N ($1 \leq N \leq 10^{100}$).

Выходные данные:

Выведите одно целое число — количество уникальных трехзначных кодов.

Ввод	Вывод
11111111111001111111	4
12	0

Решение:

```
import sys

def count_unique_three_digit_codes(access_code):
    """
    Функция подсчитывает количество уникальных трехзначных кодов,
    которые можно составить из цифр строки, удаляя некоторые из них.

    :param access_code: строка, представляющая число
    :return: количество уникальных трехзначных кодов
    """
    unique_codes = set() # Множество для хранения уникальных кодов
    length = len(access_code)
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

```
if length < 3:
    return 0 # Если длина меньше 3, невозможно составить трехзначный код

# Перебираем все возможные тройки цифр
for i in range(length - 2):
    for j in range(i + 1, length - 1):
        for k in range(j + 1, length):
            if access_code[i] != '0': # Первое число не должно быть нулем
                code = access_code[i] + access_code[j] + access_code[k]
                unique_codes.add(code)

return len(unique_codes)

access_code = sys.stdin.read().strip()
print(count_unique_three_digit_codes(access_code))
```

Задание 4. Инвентаризация мостов на острове

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

На острове «Бриджленд» расположено N деревень, и некоторые из них соединены мостами. Главный инженер острова решил провести инвентаризацию мостов, чтобы знать, сколько из них существует. Помогите ему сосчитать количество мостов на острове.

Входные данные:

В первой строке записано число N ($0 \leq N \leq 100$). В следующих N строках записано по N чисел, каждое из которых является единичкой или ноликом. Если в позиции (i, j) квадратной матрицы стоит единичка, то i -я и j -я деревни соединены мостом; если нолик — не соединены. Гарантируется, что все мосты соединяют различные деревни.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 10 класс
Вариант 2

Выходные данные:

Необходимо вывести число, определяющее количество мостов на острове «Бриджленд».

Ввод	Вывод
5 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	3

Решение:

```
import sys

def count_bridges(n, matrix):
    """
    Функция подсчитывает количество мостов на острове "Бриджленд".

    :param n: количество деревень
    :param matrix: квадратная матрица NxN, описывающая соединения мостами
    :return: количество мостов
    """
    bridges = 0 # Счетчик мостов

    # Проход по верхней треугольной матрице (без диагонали)
    for i in range(n):
        for j in range(i + 1, n):
            if matrix[i][j] == 1:
                bridges += 1

    return bridges

n = int(sys.stdin.readline().strip())
matrix = [list(map(int, sys.stdin.readline().split())) for _ in range(n)]
print(count_bridges(n, matrix))
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

Задание 1. Таинственный артефакт

Ты — молодой исследователь и программист, работающий в экспедиции по изучению древних цивилизаций. Однажды, во время раскопок в заброшенном храме, ты находишь таинственный артефакт, покрытый странными символами. Это древняя табличка, на которой изображены различные числа и буквы, напоминающие числа в разных системах счисления.

Загадка таблички заключается в том, что для расшифровки важного послания на ней, нужно преобразовать числа из одной системы счисления в другую. Эти преобразования настолько важны, что именно они могут открыть дверь в следующий зал храма, где скрыта древняя библиотека знаний.

Задача перед тобой ясна: тебе предстоит разработать программу, которая поможет конвертировать числа с одного основания в другое, и таким образом расшифровать послание, заложенное в артефакте.

Условие

Тебе предстоит написать программу, которая поможет преобразовывать числа между системами счисления. Ты получаешь число в одной системе счисления и должен перевести его в систему счисления другого основания.

Программа должна работать следующим образом:

1. На вход ты получаешь число N , представленное в системе счисления с основанием:

$$B_1 \text{ (где } 2 \leq B_1 \leq 36 \text{)}$$

2. Затем программа получает основание целевой системы счисления:

$$B_2 \text{ (где } 2 \leq B_2 \leq 36 \text{)}$$



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

Число может быть, как целым, так и дробным. Для дробных чисел программа должна сохранять точность до 3 знаков после запятой.

Формат входных данных

На вход программе подаются три строки:

- N — число в системе счисления с основанием B_1 . Оно может содержать буквы от A до Z, представляющие числа от 10 до 35, а также символ точки («.») для разделения целой и дробной части.
- B_1 — основание исходной системы счисления ($2 \leq B_1 \leq 36$).
- B_2 — основание целевой системы счисления ($2 \leq B_2 \leq 36$).

Формат выходных данных

Выведите строку, представляющую число N в системе счисления с основанием B_2 . Дробная часть должна быть округлена до 5 знаков после запятой.

Решение:

```
def convert_base(number: str, base1: int, base2: int) -> str:
    """
    Конвертирует число из одной системы счисления в другую.

    :param number: Число в виде строки в системе счисления base1.
    :param base1: Основание исходной системы счисления.
    :param base2: Основание целевой системы счисления.
    :return: Число в системе счисления base2 в виде строки.
    """
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

```
def to_decimal(number: str, base: int) -> float:
    """
    Переводит число из системы счисления с основанием base в десятичную систему.
    """
    if '.' in number:
        integer_part, fractional_part = number.split('.')
    else:
        integer_part, fractional_part = number, ''

    # Перевод целой части
    decimal_int = int(integer_part, base)
```

```
    # Перевод дробной части
    decimal_frac = 0
    for i, digit in enumerate(fractional_part, 1):
        decimal_frac += int(digit, base) / (base ** i)

    return decimal_int + decimal_frac

def from_decimal(decimal: float, base: int) -> str:
    """
    Переводит число из десятичной системы в систему с основанием base.
    """
    integer_part = int(decimal)
    fractional_part = decimal - integer_part

    # Перевод целой части
    digits = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    result_int = ''
    while integer_part > 0:
        result_int = digits[integer_part % base] + result_int
        integer_part //= base
    result_int = result_int or '0'

    # Перевод дробной части
    result_frac = []
    for _ in range(5): # До 5 знаков после запятой
        fractional_part *= base
        digit = int(fractional_part)
        result_frac.append(digits[digit])
        fractional_part -= digit
        if fractional_part == 0:
            break
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

```
        return f"{result_int}.{ ''.join(result_frac) }" if result_frac else result_int

# Переводим число в десятичную систему
decimal_number = to_decimal(number, base1)
# Переводим число из десятичной системы в целевую
return from_decimal(decimal_number, base2)

# Пример использования:
if __name__ == "__main__":
    # Чтение входных данных
    number = input()
    base1 = int(input())
    base2 = int(input())

    result = convert_base(number, base1, base2)

    # Вывод результата
    print(f"{result}")
```

Задание 2. Строка треугольника Паскаля

В древнем городе математика и гармонии жил мудрец, известный своим треугольным законом чисел. Этот закон позволял находить значение каждого элемента в треугольнике, где каждая строчка строилась на основе двух чисел, расположенных прямо над ней. Однажды ученик спросил у мудреца: «Как найти конкретную строку треугольника, если известен её номер?» Мудрец ответил: «Каждое число в строке — это сумма двух чисел, стоящих над ним. Начиная с вершины, и ты придёшь к ответу.» Теперь задача найти конкретную строку треугольника Паскаля.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

Формат ввода:

Одно целое число **rowIndex** ($0 \leq \text{rowIndex} \leq 33$) — номер строки треугольника Паскаля, который необходимо найти. Нумерация начинается с 0.

Формат вывода:

Список из **rowIndex+1** элементов, представляющий строку треугольника Паскаля.

Решение:

```
def generate_row(row_index):  
    """  
    Генерирует строку Паскаля для заданного индекса строки.  
  
    Строка Паскаля — это последовательность коэффициентов бинома Ньютона. Каждое число в  
    строке  
    вычисляется как произведение предыдущего числа на отношение чисел, соответствующих  
    порядку и индексу.  
  
    Аргументы:  
    row_index -- индекс строки Паскаля (целое число)  
  
    Возвращает:  
    list -- строка Паскаля в виде списка чисел  
    """  
    # Начальная строка (первая строка всегда состоит из одного числа 1)  
    row = [1]  
  
    # Генерация коэффициентов для строки Паскаля  
    for i in range(1, row_index + 1):  
        # Каждое следующее число вычисляется по формуле: предыдущий коэффициент *  
        (row_index - i + 1) / i  
        row.append(row[i - 1] * (row_index - i + 1) // i)  
  
    return row  
  
if __name__ == '__main__':  
    """  
    Вводит индекс строки Паскаля, вызывает функцию для её генерации и выводит результат.  
    """  
    # Вводим индекс строки  
    row_index = int(input())
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

```
# Генерация строки и вывод результата  
print(" ".join(map(str, generate_row(row_index))))
```

Задание 3. Восстановление валидных IP-адресов

Представьте, что вы инженер-программист, разрабатывающий сеть для большого корпоративного офиса. Вам дали длинную строку, содержащую цифры, которые представляют IP-адреса в зашифрованном виде (без точек). Вам нужно быстро восстановить все возможные валидные IP-адреса, которые могут быть получены из этой строки, чтобы протестировать различные сценарии маршрутизации. Важно, чтобы все результаты выводились упорядоченными по возрастанию, чтобы легко проверить диапазоны.

Формат ввода:

На вход подаётся строка s , состоящая из цифр (0–9), длиной от 1 до 20 символов.

- $1 \leq \text{len}(s) \leq 20$
- Строка не содержит пробелов или других символов.

Формат вывода:

Выводятся все возможные валидные IP-адреса, которые можно составить из строки, по одному на каждой строке.

IP-адреса выводятся в лексикографическом порядке (по возрастанию).



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

Решение:

```
def restore_ip_addresses(s: str):  
    """  
    Восстанавливает все возможные корректные IP-адреса из заданной строки.  
  
    Каждый IP-адрес состоит из 4 числовых сегментов, разделенных точками. Каждый сегмент  
    должен быть числом от 0 до 255,  
    и не должен иметь ведущих нулей, если его длина больше 1.  
  
    Аргументы:  
    s -- строка, представляющая числа, из которых нужно восстановить IP-адреса  
  
    Возвращает:  
    list -- список строк с возможными корректными IP-адресами  
    """  
    def is_valid(segment: str) -> bool:  
        """  
        Проверяет, является ли строка допустимым сегментом для IP-адреса.  
  
        Сегмент считается допустимым, если:  
        1. Его длина не более 3 символов.  
        2. Он не имеет ведущих нулей (если длина больше 1).  
        3. Его значение в пределах от 0 до 255.  
  
        Аргументы:  
        segment -- строка, представляющая сегмент IP-адреса  
  
        Возвращает:  
        bool -- True, если сегмент допустим, иначе False  
        """  
        if len(segment) > 1 and segment[0] == '0': # Проверка на ведущие нули  
            return False  
        if 0 <= int(segment) <= 255:  
            return True  
        return False  
  
    def backtrack(start: int, path: list[str]):  
        """  
        Рекурсивно генерирует все возможные IP-адреса с использованием метода обратного  
        хода.
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

```
Аргументы:
start -- индекс в строке, с которого начинается текущая подстрока
path -- текущий список сегментов, который строится для потенциального IP-адреса
"""

if start == len(s) and len(path) == 4: # Если строка обработана и есть 4 сегмента
    valid_ips.append(".".join(path)) # Добавляем сформированный IP-адрес в список
    return
if len(path) >= 4: # Если уже есть 4 сегмента, завершаем обработку
    return

for length in range(1, 4): # Пробуем сегменты длиной от 1 до 3
    if start + length <= len(s):
        segment = s[start:start + length]
        if is_valid(segment): # Если сегмент допустим
            backtrack(start + length, path + [segment]) # Рекурсивно добавляем
следующий сегмент

valid_ips = [] # Список для хранения валидных IP-адресов
backtrack(0, []) # Начинаем с первого индекса и пустого пути
valid_ips.sort() # Сортируем IP-адреса по возрастанию

return valid_ips

if __name__ == '__main__':
    """
    Вводит строку, вызывает функцию для восстановления IP-адресов и выводит результат.
    """
    s = input()
    valid_ips = restore_ip_addresses(s)
    print("\n".join(valid_ips))
```

Задание 4. Сбор дождевой воды

После долгой засухи, в вашем городе наконец-то прошел долгожданный дождь. Вода начала собираться в низинах между зданиями, высота которых известна заранее. Теперь вам поручили задачу: вычислить, сколько воды удалось собрать в этих «лужах» между зданиями, чтобы использовать её для нужд города.

Здания выстроены в одну линию, и их высоты представлены массивом, где каждое число обозначает высоту здания в этой позиции. Ваша задача —



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

рассчитать общий объём воды, который можно собрать между этими зданиями, считая, что дно идеально ровное, а ширина каждого здания и промежутка между ними равна единице.

Формат ввода:

- Первая строка содержит одно целое число n ($1 \leq n \leq 10^4$) — количество элементов в массиве.
- Вторая строка содержит n неотрицательных целых чисел $height[i]$ ($1 \leq height[i] \leq 10^4$), разделённых пробелами, — высоты столбцов.

Формат вывода:

- Одно целое число — общий объём воды, который может быть собран между столбцами.

Решение:

```
def calculate_water_trapped(height):  
    """  
    Вычисляет количество воды, которое может быть задержано между столбиками на основе их  
    высоты.  
  
    Используется алгоритм с двумя массивами (left_max и right_max), которые хранят  
    максимальные высоты  
    столбиков слева и справа от каждого столбика. Затем для каждого столбика определяется,  
    сколько воды  
    может быть удержано, исходя из минимального значения между left_max и right_max,  
    вычитая высоту столбика.  
  
    Аргументы:  
    height — список целых чисел, представляющих высоту столбиков
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс

```
Возвращает:
integer -- количество воды, которое может быть задержано
"""

# Если список пуст, вода не может быть задержана
if not height:
    return 0

n = len(height)

# Массивы для хранения максимальных высот слева и справа
left_max = [0] * n
right_max = [0] * n
water = 0

# Заполнение left_max (максимальные высоты слева)
left_max[0] = height[0]
for i in range(1, n):
    left_max[i] = max(left_max[i - 1], height[i])

# Заполнение right_max (максимальные высоты справа)
right_max[n - 1] = height[n - 1]
for i in range(n - 2, -1, -1):
    right_max[i] = max(right_max[i + 1], height[i])

# Подсчёт воды для каждого столбика
for i in range(n):
    water += max(0, min(left_max[i], right_max[i]) - height[i])

return water

if __name__ == '__main__':
    """
    Вводит список высот столбиков, вызывает функцию для вычисления количества
    задерживаемой воды
    и выводит результат.
    """
    # Ввод данных: строка, представляющая высоты столбиков
    n = int(input())
    height = list(map(int, input().split()))

    # Вывод результата
    print(calculate_water_trapped(height))
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

Задание 1. Количество пропеллеров на самолете

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

В авиапарке строится новый экспериментальный самолет с множеством пропеллеров. Каждый пропеллер имеет определенное количество лопастей в зависимости от номера детали: цифра 0 обозначает пропеллер с 1 лопастью, цифра 6 — пропеллер с 1 лопастью, цифра 8 — пропеллер с 2 лопастями и т.д. Напишите программу, которая посчитает общее количество лопастей на всех пропеллерах самолета по номеру самолета.

Входные данные:

Вам дано целое число N ($0 \leq N \leq 10^{100}$) — номер самолета.

Выходные данные:

Выведите одно число — количество лопастей на всех пропеллерах самолета.

Ввод	Вывод
157892	3
409578	4

Решение:

```
def count_blades(number_str):  
    """  
    Подсчитывает общее количество лопастей, присутствующих на цифрах числа.  
  
    Каждая цифра на цифровом дисплее может содержать разное количество лопастей.  
    Эта функция суммирует количество лопастей для каждой цифры в строке.  
  
    Аргументы:  
    number_str -- строка, представляющая число (состоящая из цифр от '0' до '9')  
    """
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

```
Возвращает:
total_blades -- общее количество лопастей для всех цифр
"""

# МAPPING: цифра -> количество лопастей
blades_count = {
    '0': 1,
    '1': 0,
    '2': 0,
    '3': 0,
    '4': 1,
    '5': 0,
    '6': 1,
    '7': 0,
    '8': 2,
    '9': 1
}

# Считаем количество лопастей по каждой цифре
total_blades = sum(blades_count[digit] for digit in number_str)

return total_blades

if __name__ == '__main__':
    """
    Вводит строку с числом, вызывает функцию для подсчета лопастей
    и выводит результат.
    """
    number_str = input()
    print(count_blades(number_str))
```

Задание 2. Фруктовый рынок

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

На фруктовом рынке в рядах стояли три корзины с фруктами: апельсины, яблоки и бананы. Каждый день утром хозяин рынка менял местами стоящую справа корзину с центральной. А вечером помощник менял местами левую корзину с центральной. Требуется определить порядок корзин через K дней.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

Входные данные:

Дано одно натуральное число K — количество дней ($1 \leq K \leq 1000$).

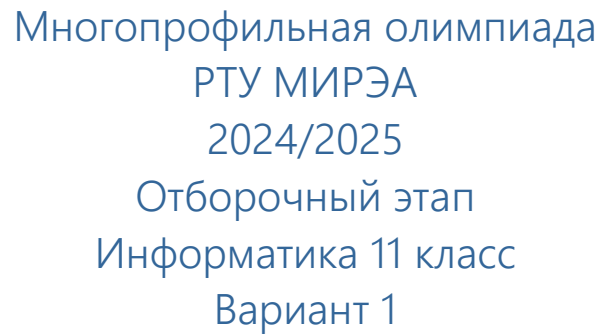
Выходные данные:

Ответ должен содержать три заглавные буквы: «O», «A» и «B» (без пробелов), описывающие порядок фруктов слева направо по прошествии K дней. Обозначения: O — апельсины (oranges), A — яблоки (apples), B — бананы (bananas).

Ввод	Вывод
3	OAB
2	ABO

Решение:

```
def find_fruit_order(K):  
    """  
    Находит порядок фруктов в корзинах по индексу дня.  
  
    Порядок корзин циклический и повторяется каждые 3 дня.  
    Порядок корзин для трех дней: 'OAB', 'BOA', 'ABO'.  
  
    Аргументы:  
    K — номер дня (целое число, начиная с 0)  
  
    Возвращает:  
    string — порядок фруктов на день K  
    """  
    # Исходный порядок корзин: O (апельсины), A (яблоки), B (бананы)  
    orders = ['OAB', 'BOA', 'ABO'] # Циклический порядок  
    # Порядок повторяется каждые 3 дня  
    return orders[K % 3]  
  
if __name__ == '__main__':  
    """  
    Вводит номер дня и выводит порядок фруктов в корзинах на этот день.  
    """  
    K = int(input())  
    print(find_fruit_order(K))
```



Ограничение по памяти: 16 МВ

Выведите матрицу, заполненную числами от 1 до N^2 по диагоналям, начиная с верхнего левого угла. Между числами пробел.

Ввод	Вывод
4	1 2 6 7 3 5 8 13 4 9 12 14 10 11 15 16



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

Решение:

```
def solution_task(n):  
    """  
    Заполняет квадратную матрицу размером n x n по определенному пути и возвращает  
    строковое представление этой матрицы.
```

Путь заполнения:

1. Стартует с верхнего левого угла.
2. Направление движения по матрице меняется по циклическому паттерну.
3. В случае достижения границы матрицы, направление меняется.

Аргументы:

n -- размерность матрицы (целое число)

Возвращает:

string -- строковое представление матрицы с числами, разделенными пробелами
"""

Направления движения по матрице: (x, y) – шаги по оси x и y

```
direction = [(1, 0), (-1, 1), (0, 1), (1, -1)]
```

Создаем пустую матрицу n x n, заполненную нулями

```
matrix = [[0] * n for _ in range(n)]
```

```
num = 1 # Начальное число для заполнения
```

```
x, y = 0, 0 # Начальная позиция в верхнем левом углу
```

```
step = 0 # Индикатор текущего шага (для выбора направления)
```

```
while num <= n * n:
```

```
    # Записываем текущее число в матрицу
```

```
    matrix[x][y] = num
```

```
    num += 1 # Переходим к следующему числу
```

```
    # Обработка шага по направлению
```

```
    if step % 2 == 0:
```

```
        # Если шаг четный, следуем по направлению в пределах массива
```

```
        y += direction[step][0]
```

```
        x += direction[step][1]
```

```
        step += 1
```

```
    else:
```

```
        # Если шаг нечетный, проверяем границу матрицы
```

```
        y += direction[step][0]
```

```
        x += direction[step][1]
```

```
        # Если мы достигли границы, меняем направление
```

```
        if x == n - 1 or y == n - 1 or x == 0 or y == 0:
```

```
            step += 1
```



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

```
# Корректируем координаты, если мы вышли за пределы матрицы
if x == n:
    y += 1
    x -= 1
if y == n:
    x += 1
    y -= 1

# Обновляем шаг, чтобы он циклично повторялся
step %= 4

# Формируем строковое представление матрицы с числами
return '\n'.join(map(' '.join, map(lambda x: map(str, x), matrix)))

if __name__ == '__main__':
    """
    Вводит размер матрицы, вызывает функцию для решения задачи и выводит результат.
    """
    n = int(input())
    print(solution_task(n))
```

Задание 4. Распределение ролей на спектакле

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

В театральной труппе участвуют N актеров. Для спектакля требуется выбрать трех актеров на роли главного героя, антагониста и комического персонажа. Сколько существует вариантов распределения актеров на эти три роли, если каждую роль может сыграть только один актер?

Входные данные:

Вам дано целое число N ($1 \leq N \leq 104$) — количество актеров в труппе.



Многопрофильная олимпиада
РТУ МИРЭА
2024/2025
Отборочный этап
Информатика 11 класс
Вариант 1

Выходные данные:

Выведите единственное число — количество способов распределения актеров на три роли.

Ввод	Вывод
5	60
3	6

Решение:

```
def count_ways_to_assign_actors(n):  
    """  
    Вычисляет количество способов распределения актеров для роли, если необходимо выбрать  
    3 актера из n доступных, учитывая порядок.  
  
    Аргументы:  
    n — количество доступных актеров (целое число)  
  
    Возвращает:  
    integer — количество способов распределения актеров  
    """  
    if n < 3:  
        # Если актеров меньше 3, способов распределить их нет  
        return 0  
    else:  
        # Если актеров 3 или больше, то количество способов — это  $n * (n-1) * (n-2)$   
        return n * (n - 1) * (n - 2)  
  
if __name__ == '__main__':  
    """  
    Вводит количество актеров, вызывает функцию для вычисления способов распределения  
    актеров и выводит результат.  
    """  
    n = int(input())  
    print(count_ways_to_assign_actors(n))
```



Многопрофильная олимпиада

РТУ МИРЭА

2024/2025

Отборочный этап

Информатика 11 класс

Вариант 2

Задание 1. Учёт книг и полок в библиотеке

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

После проведения переезда в новую библиотеку было обнаружено, что при транспортировке утерялись как книги, так и полки. При этом полка, на которой нет книги, или книга без полки считается непригодной для использования. Напишите программу, которая определит общий ущерб для библиотеки.

Входные данные:

На первой строке записаны три целых числа: b_1 , p_1 , c_1 — начальное количество книг ($100 \leq b_1 \leq 30000$, b_1 кратно 100), процент потерянных книг ($0 \leq p_1 \leq 100$) и стоимость одной книги ($1 \leq c_1 \leq 100$). Вторая строка содержит три целых числа: s_1 , p_2 , c_2 — начальное количество полок ($100 \leq s_1 \leq 30000$, s_1 кратно 100), процент потерянных полок ($0 \leq p_2 \leq 100$) и стоимость одной полки ($1 \leq c_2 \leq 100$).

Выходные данные:

Выведите одно целое число — общий ущерб библиотеки.

Ввод	Вывод
1000 10 100 1200 20 90	37000
5000 15 23 4000 17 22	53600



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

Решение:

```
def calculate_damage(b1, p1, c1, s1, p2, c2):  
    """  
    Вычисляет общий ущерб, учитывая потерю книг, полок и оставшиеся книги и полки.  
  
    Формула для расчета ущерба:  
    1. Потери книг и полок вычисляются по проценту.  
    2. Ущерб от потерянных книг и полок рассчитывается как количество утраченных  
    элементов, умноженное на их стоимость.  
    3. Ущерб от оставшихся книг без полок и полок без книг рассчитывается, если  
    соответствующие элементы превышают друг друга.  
  
    Аргументы:  
    b1 -- начальное количество книг (целое число)  
    p1 -- процент потерянных книг (целое число)  
    c1 -- стоимость одной книги (целое число)  
    s1 -- начальное количество полок (целое число)  
    p2 -- процент потерянных полок (целое число)  
    c2 -- стоимость одной полки (целое число)  
  
    Возвращает:  
    total_damage -- общий ущерб (целое число)  
    """  
    # Вычисление потерянных книг и полок  
  
    lost_books = (b1 * p1) // 100  
    lost_shelves = (s1 * p2) // 100  
  
    # Оставшиеся книги и полки  
    remaining_books = b1 - lost_books  
    remaining_shelves = s1 - lost_shelves  
  
    # Ущерб от потерянных книг и полок  
    damage_from_lost_books = lost_books * c1  
    damage_from_lost_shelves = lost_shelves * c2  
  
    # Ущерб от оставшихся книг без полок  
    if remaining_books > remaining_shelves:  
        damage_from_books_without_shelves = (remaining_books - remaining_shelves) * c1  
    else:  
        damage_from_books_without_shelves = 0  
  
    # Ущерб от оставшихся полок без книг  
    if remaining_shelves > remaining_books:  
        damage_from_shelves_without_books = (remaining_shelves - remaining_books) * c2  
    else:  
        damage_from_shelves_without_books = 0
```



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

```
# Общий ущерб
total_damage = (damage_from_lost_books +
                damage_from_lost_shelves +
                damage_from_books_without_shelves +
                damage_from_shelves_without_books)

return total_damage

if __name__ == '__main__':
    """
    Вводит данные и вызывает функцию для расчета ущерба.
    """
    # Ввод данных
    b1, p1, c1 = map(int, input().split())
    s1, p2, c2 = map(int, input().split())

    # Выводим результат
    print(calculate_damage(b1, p1, c1, s1, p2, c2))
```

Задание 2. Исправление пароля

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

Алекс получает доступ к своему компьютеру через пароль, состоящий из заглавных английских букв. Однако иногда он случайно вводит лишнюю букву, которая сбивает систему. Чтобы ему не приходилось вводить пароль заново, он хочет написать программу, которая удаляет одну лишнюю букву и выдает исправленный пароль.

Входные данные:

В строке содержится целое число K — номер лишней буквы, а затем через пробел записан пароль S , состоящий из заглавных английских букв. Гарантируется, что K от 1 до длины пароля включительно. Длина пароля не более 80 символов. Нумерация начинается с 1.



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

Выходные данные:

Выведите исправленный пароль.

Ввод	Вывод
4 PASSWORD	PASSWORD
6 COMPUTER	COMPUTER

Решение:

```
def solution_task(k, password):  
    """  
    Удаляет символ из пароля по указанному индексу.  
  
    Функция принимает индекс символа (k) и строку пароля, после чего удаляет символ  
    на позиции k и возвращает обновленный пароль.  
  
    Аргументы:  
    k — индекс символа, который нужно удалить (целое число в строковом виде)  
    password — строка пароля, из которой нужно удалить символ  
  
    Возвращает:  
    string — обновленная строка пароля после удаления символа  
    """  
    # Преобразуем строку пароля в список, чтобы можно было удалить элемент по индексу  
    password = list(password)  
  
    # Удаляем символ по индексу k (считаем, что k задан с 1)  
    password.pop(int(k) - 1)  
  
    # Преобразуем список обратно в строку и возвращаем результат  
    return ''.join(password)  
  
if __name__ == '__main__':  
    """  
    Вводит индекс и пароль, вызывает функцию для удаления символа и выводит результат.  
    """  
    print(solution_task(*input().split()))
```



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

Задание 3. Заполнение зигзагом

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

Требуется заполнить квадратную матрицу размером $N \times N$ числами от 1 до N^2 по зигзагообразной траектории: первая строка заполняется слева направо, вторая — справа налево, третья — снова слева направо, и так далее.

Входные данные:

Задано целое число N ($2 \leq N \leq 100$) — размер квадратной матрицы.

Выходные данные:

Выведите матрицу, заполненную числами от 1 до N^2 зигзагообразно. Между числами может быть любое количество пробелов.

Ввод	Вывод
4	1 2 3 4 8 7 6 5 9 10 11 12 16 15 14 13

Решение:

```
def solution_task(n):  
    """  
    Заполняет квадратную матрицу размером n x n числами от 1 до n*n по зигзагообразному  
    пути.  
  
    Числа заполняются по следующему принципу:  
    1. Начинаем с верхнего левого угла.  
    2. Каждое следующее число размещается по горизонтали, причем направление по очереди  
    меняется  
    между движением вправо и влево через каждую строку.  
    3. Если мы достигаем конца строки, переходим на следующую строку, меняя направление.  
  
    Аргументы:  
    n — размерность матрицы (целое число)
```



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

```
Возвращает:
string -- строковое представление заполненной матрицы, где числа разделены пробелами
"""

# Инициализация пустой матрицы n x n, заполненной нулями
matrix = [[0] * n for _ in range(n)]

num = 1 # Начальное число для заполнения
x, y = 0, 0 # Начальная позиция в верхнем левом углу

# Описание направления для движения по столбцам (вправо / влево)
description = [1, -1]

# Заполняем матрицу
while num <= n * n:
    matrix[x][y] = num # Размещаем число в текущей ячейке
    num += 1 # Переходим к следующему числу

    # Двигаемся по текущему направлению (вправо или влево)
    y += description[x % 2]

    # Если мы достигли конца строки, переходим на следующую
    if (num - 1) % n == 0:
        x += 1
        # Если строка нечетная, двигаемся влево, иначе вправо
        if x % 2:
            y -= 1
        else:
            y += 1

# Формируем строковое представление матрицы с числами, разделенными пробелами
return '\n'.join(map(' '.join, map(lambda x: map(str, x), matrix)))

if __name__ == '__main__':

    """
    Вводит размер матрицы, вызывает функцию для решения задачи и выводит результат.
    """

    # Ввод данных и вывод результата
    print(solution_task(int(input())))
```



Многопрофильная олимпиада

PTU MIREA

2024/2025

Отборочный этап

Информатика 11 класс

Вариант 2

Задание 4. Подбор команды программистов

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 MB

Миша собирает команду программистов для участия в хакатоне. Ему нужно выбрать одного разработчика, одного дизайнера и одного менеджера. Известно, что у него есть R разработчиков, D дизайнеров и M менеджеров. Сколько существует различных способов сформировать такую команду?

Входные данные:

В строке содержится три целых числа R , D и M ($1 \leq R, D, M \leq 1000$) — количество разработчиков, дизайнеров и менеджеров соответственно.

Выходные данные:

Выведите одно целое число — количество способов формирования команды.

Ввод	Вывод
3 4 5	60
2 3 1	6



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

Решение:

```
def solution_task(a, b, c):  
    """  
    Вычисляет произведение трех чисел.  
  
    Аргументы:  
    a -- первое число (целое число)  
    b -- второе число (целое число)  
    c -- третье число (целое число)  
  
    Возвращает:  
    integer -- произведение чисел a, b и c  
    """  
    return a * b * c  
  
if __name__ == '__main__':  
    """  
    Вводит три числа, вызывает функцию для вычисления их произведения и выводит результат.  
    """  
    # Ввод данных: три числа через пробел  
    print(solution_task(*map(int, input().split())))
```