

Вариант олимпиады по ИИ для 7–8 классов

Задача 1. Логотипы кружка ИИ

Маша делает логотипы для кружка ИИ. У неё есть:

- F вариантов **фона** (пронумерованы от 1 до F),
- S вариантов **символа**,
- C вариантов **цвета текста** (пронумерованы от 1 до C).

Из соображений читаемости **цвет текста не должен совпадать с фоном**:

если выбран фон №i, то нельзя выбрать цвет №i. Остальные сочетания допустимы.

Задача.

Найдите количество допустимых логотипов.

Формат ввода

F S C

$1 \leq F, S, C \leq 10^9$

Формат вывода

Одно целое число — количество возможных логотипов.

Пример

Ввод:

2 3 2

Вывод:

6

ТЕСТЫ

Ввод:

1 1 1

Вывод:

0

Ввод:

3 4 3

Вывод:

24

Ввод:

5 1 5

Вывод:

20

Ввод:

1000000000 1000000000 1000000000

Вывод:

9999999990000000000000000000

Решение задачи 1

Всего $F \cdot S \cdot C$ логотипов, запрещены пары (фон=i, цвет=i)

Ответ = $S \cdot (F \cdot C - \min(F, C))$

```
def main():
```

```
    F, S, C = map(int, input().split())
```

```
    forbidden = min(F, C)
```

```
    ans = S * (F * C - forbidden)
```

```
    print(ans)
```

```
if __name__ == "__main__":
```

```
    main()
```

Задача 2. Монетка библиотекаря

Робот-библиотекарь выбирает монетку: с вероятностью p берётся **фальшивая**, которая всегда даёт «орёл», а с вероятностью $1 - p$ — **честная**, у которой «орёл» и «решка» равновероятны. Монетку подбросили один раз, и выпал «орёл». Какова вероятность, что выбрана фальшивая монетка?

Дано рациональное p как несократимая дробь $\frac{a}{b}$. Выведите искомую вероятность в виде **несократимой дроби** $\frac{num}{den}$.

Подсказка: используйте формулу Байеса:

$$P(F | H) = \frac{P(H | F) P(F)}{P(H | F) P(F) + P(H | C) P(C)} = \frac{1 \cdot p}{1 \cdot p + 1/2 \cdot (1 - p)}$$

Формат ввода.

Одна строка: два целых числа a и b ($1 \leq a \leq b \leq 10^9$, $\gcd(a, b) = 1$), задающих $p = \frac{a}{b}$.

Формат вывода.

Два целых числа num den — числитель и знаменатель несократимой дроби.

Пример 1

Ввод

1 2

Вывод

2 3

ТЕСТЫ

Ввод:

1 1

Вывод:

1 1

Ввод:

1 3

Вывод:

1 2

Ввод:

2 3

Вывод:

4 5

Ввод:

2 6

Вывод:

1 2

Решение задачи 2

$P(F|H) = 2p / (1 + p)$; при $p=a/b \rightarrow 2a / (a + b)$, сократить дробь.

```
import math
```

```
def main():
```

```
    a, b = map(int, input().split())
```

```
    num = 2 * a
```

```
    den = a + b
```

```
    g = math.gcd(num, den)
```

```
    print(num // g, den // g)
```

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

Задача 3. Оценки без «пустых» (анализ данных)

После мини-контрольной по Python наставник выгрузил оценки в систему. Некоторые записи оказались пустыми — отмечены как NA. Для отчёта нужны **среднее** и **медиана** только по валидным числам.

Игнорируя NA, вычислите среднее (с точностью до 1 знака после запятой, обычное округление) и медиану (если чётное количество валидных значений — среднее двух центральных, также до 1 знака).

Формат ввода.

Первая строка: целое n ($1 \leq n \leq 10^5$).

Вторая строка: n токенов — либо целые числа $[-10^9; 10^9]$, либо строка NA.

Формат вывода.

Два числа: среднее и медиана с точностью до 1 знака.

Если нет ни одного валидного значения — выведите NA NA.

Пример

Ввод

```
5  
3 5 NA 4 5
```

Вывод

```
4.2 4.5
```

Тесты

Ввод:

```
3  
1 2 3
```

Вывод:

```
2.0 2.0
```

Ввод:

```
4  
NA NA NA NA
```

Вывод:

```
NA NA
```

Ввод:

4

1 2 3 4

Вывод:

2.5 2.5

Ввод:

6

10 NA 5 NA 0 5

Вывод:

5.0 5.0

Решение задачи 3

```
def main():
```

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

```
    tokens = input().split()
```

```
    vals = [int(t) for t in tokens if t != 'NA']
```

```
    if not vals:
```

```
        print("NA NA")
```

```
        return
```

```
    vals.sort()
```

```
    m = len(vals)
```

```
    mean = sum(vals) / m
```

```
    if m % 2 == 1:
```

```
        med = vals[m // 2]
```

```
    else:
```

```
        med = (vals[m // 2 - 1] + vals[m // 2]) / 2
```

```
    mean = round(mean, 1)
```

```
    med = round(med, 1)
```

```
print(f'{mean:.1f} {med:.1f}')
```

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

Задача 4. Прогулка робота по карте

Учебный робот перемещается по клетчатой плоскости, начиная из (0,0)(0,0).
Команды:

- N — $y := y + 1$
- S — $y := y - 1$
- E — $x := x + 1$
- W — $x := x - 1$

Робот иногда получает повторы команд в виде «число + буква», например 3N — три шага на север. Без числа — один шаг.

Задача. По строке команд найдите конечные координаты (x, y) .

Формат ввода.

Одна строка — непустая команда длиной до 10^5 , состоящая из цифр и букв NSEW.
Числа — положительные, без ведущих нулей, не превышают 10^9 .

Формат вывода.

Два целых числа x и y .

Пример 1

Ввод

NNEESW

Вывод

1 1

Тесты

Ввод:

2SW3E6S

Вывод:

2 -8

Ввод:

2N10E3S

Вывод:

10 -1

Ввод:

W

Вывод:

-1 0

Ввод:

3E2S3W2N

Вывод:

0 0

Решение задачи 4

```
def main():  
    s = input().strip()  
    x = y = 0  
    num = 0  
    have_num = False  
  
    for ch in s:  
        if ch.isdigit():  
            num = num * 10 + int(ch)  
            have_num = True  
        else:  
            k = num if have_num else 1  
            if ch == 'N':  
                y += k  
            elif ch == 'S':  
                y -= k  
            elif ch == 'E':  
                x += k  
            elif ch == 'W':  
                x -= k  
            num = 0  
            have_num = False
```

```
print(x, y)
```

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

Задача 5. Антишум: чистим сообщение

Голосовой ассистент распознал текст, но для компактной подписи нужно убрать все **гласные** (и латинские, и русские), а также все пробелы подряд сократить до одного.

Считаем гласными: AEIOUYaeiouyAEЁIOУЫЭЮЯaeёioуыэюя.

Удалите все гласные. Любые последовательные пробелы в результате замените на один пробел. Начальные и конечные пробелы уберите.

Формат ввода.

Одна строка произвольного текста (длина $\leq 10^5$).

Формат вывода.

Одна строка — «очищенный» текст.

Пример

Ввод

Artificial Intelligence

Вывод

rtfcl ntlgnc

Тесты

Ввод:

Hello, World!

Вывод:

Hll, Wrld!

Ввод:

Привет, это олимпиада по ИИ!

Вывод:

Првт, т лмпд п !

Ввод:
AEIOUY
Вывод:

Ввод:
Hello World
Вывод:
Hll Wrld

Решение задачи 5

```
VOWELS = set("AEIOUYaeiouyAEЁИОУЫЭЮЯaeёиоуыэюя")
```

```
def main():  
    s = input()  
    filtered = ''.join(ch for ch in s if ch not in VOWELS)  
    result = ' '.join(filtered.split())  
    print(result)  
  
if __name__ == "__main__":  
    main()
```

Задача 6. Две команды

Учитель делит участников на две команды для хакатона: в «Альфа» попадают все, у кого балл **не ниже среднего**, в «Бета» — остальные. Внутри каждой команды порядок нужно сохранить как во входе. Разбейте список оценок на две группы по правилу выше.

Формат ввода.

Первая строка: n ($1 \leq n \leq 2 \cdot 10^5$).

Вторая строка: n целых чисел $[-10^9; 10^9]$.

Формат вывода.

Две строки.

1. Баллы команды «Альфа» (через пробел) — те, что $\geq$ среднего арифметического по всем n .
2. Баллы команды «Бета».

Если команда пуста — выведите пустую строку.

Пример

Ввод

```
6
10 5 8 7 9 3
```

Вывод

```
10 8 7 9
5 3
```

Тесты

Ввод:

```
5
5 5 5 5 5
```

Вывод:

```
5 5 5 5 5
```

Ввод:

```
4
1 2 3 10
```

Вывод:

```
10
1 2 3
```

Ввод:

```
3
1 2 3
```

Вывод:

```
2 3
1
```

Ввод:

```
5
-5 -1 -3 -4 -2
```

Вывод:

```
-1 -3 -2
-5 -4
```

Решение задачи 6

```
def main():
```

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

```
arr = list(map(int, input().split()))
total = sum(arr)

alpha, beta = [], []

for v in arr:
    if v * n >= total:
        alpha.append(v)
    else:
        beta.append(v)

print(*alpha)
print(*beta)

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

Задача 7. Сколько вариантов бросков?

Симуляция игры: игрок бросает **две стандартные кости** (от 1 до 6 на каждой). Нас интересует число исходов, где сумма значений не меньше заданного порога T .

Посчитайте количество упорядоченных пар (i, j) , $1 \leq i, j \leq 6$, таких что $i + j \geq T$.

Формат ввода.

Одно целое T ($2 \leq T \leq 12$).

Формат вывода.

Одно целое — количество исходов.

Пример

Ввод

2

Вывод

Тесты

Ввод:

7

Вывод:

21

Ввод:

12

Вывод:

1

Ввод:

3

Вывод:

35

Ввод:

11

Вывод:

3

Решение задачи 7

```
def main():
```

```
    T = int(input())
```

```
    count = 0
```

```
    for i in range(1, 7):
```

```
        for j in range(1, 7):
```

```
            if i + j >= T:
```

```
                count += 1
```

```
    print(count)
```

```
if __name__ == "__main__":
```

```
    main()
```

Задача 8. Маленькая «нейросеть»: переходы по шагам

У нас есть двухсостоятая система (например, «пассивен/активен»). Переходы за один шаг заданы матрицей 2×2 целых чисел A . Текущее состояние — целочисленный вектор v размера 2. Нужно узнать состояние через k шагов: $v_k = A^k \cdot v$.

Вычислите v_k . Числа могут вырасти — используйте длинную арифметику (в Python — `int`, в C++ — `long` хватит при ограничениях ниже).

Формат ввода.

Пять строк:

1. целое k ($0 \leq k \leq 10^9$)
2. два целых — первая строка матрицы AA
3. два целых — вторая строка матрицы AA
4. одно целое — v_1
5. одно целое — v_2

Формат вывода.

Два целых — компоненты v_k .

Пример

Ввод

```
2
1 1
1 0
1
0
```

Вывод

```
2 1
```

Тесты

Ввод:

```
0
2 3
4 5
7
11
```

Вывод:

```
7 11
```

Ввод:

```
1
1 0
```

```
0 1
5
6
Вывод:
5 6
```

```
Ввод:
3
2 0
0 2
1
1
Вывод:
8 8
```

```
Ввод:
5
1 1
1 1
1
0
Вывод:
16 16
```

(При $k = 0$ считаем A^0 тождественной матрицей, $v_0 = v$.)

Ограничения. Эффективное решение — бинарное возведение матрицы в степень за $O(\log k)$

Решение задачи 8

```
def mat_mul(A, B):
    return [
        [A[0][0]*B[0][0] + A[0][1]*B[1][0],
         A[0][0]*B[0][1] + A[0][1]*B[1][1]],
        [A[1][0]*B[0][0] + A[1][1]*B[1][0],
         A[1][0]*B[0][1] + A[1][1]*B[1][1]],
    ]
```

```
def mat_pow(A, k):
    R = [[1, 0], [0, 1]]
    while k > 0:
```

```

    if k % 2 == 1:
        R = mat_mul(R, A)
    A = mat_mul(A, A)
    k //= 2
return R

def mat_vec_mul(A, v):
    return [A[0][0]*v[0] + A[0][1]*v[1],
            A[1][0]*v[0] + A[1][1]*v[1]]

def main():
    k = int(input())
    a11, a12 = map(int, input().split())
    a21, a22 = map(int, input().split())
    v1 = int(input())
    v2 = int(input())

    A = [[a11, a12], [a21, a22]]
    if k == 0:
        print(v1, v2)
        return

    Ak = mat_pow(A, k)
    res = mat_vec_mul(Ak, [v1, v2])
    print(res[0], res[1])

if __name__ == "__main__":
    main()

```