

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Задача: Идеальное Расписание**

Ограничение времени 1 секунда

Ограничение памяти 64.0 Мб

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

В школе есть  $N$  классов ( $1 \leq N \leq 100$ ), и школа хочет ввести новую систему расписания. Каждый класс должен посетить  $M$  различных уроков в день ( $1 \leq M \leq 100$ ). У каждого класса есть свой список предпочтений, в котором указаны уроки, которые класс хочет посетить и в какой последовательности. Ваша задача - написать программу, которая определяет, возможно ли составить расписание таким образом, чтобы все классы посещали уроки в точном соответствии с их предпочтениями. Если это возможно, программа должна вывести 1, в противном случае - 0.

**Формат ввода**

Первая строка содержит два целых числа  $N$  и  $M$ . Каждая из следующих  $N$  строк содержит  $M$  целых чисел, представляющих уроки в порядке предпочтений для каждого класса.

**Формат вывода**

Программа должна вывести 1, если возможно составить такое расписание, чтобы все классы следовали своему идеальному порядку уроков, и 0, если это невозможно.

**Пример**

**Ввод**   **Вывод**

```
3 3
1 2 3
3 1 2
2 3 1
```

**Примечания**

В примере возможно устроить так, что каждый класс посещает уроки в желаемом порядке. Первый класс посещает уроки в порядке 1, 2, 3, второй класс - в порядке 3, 1, 2, и третий класс - в порядке 2, 3, 1.

**Пример решения**

```
def prt(n, m, a):
    for i in range(m):
        l = []
        for j in range(n):
            l.append(a[j][i])

        if len(set(l)) != n:
            print(0)
            return
```

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ  
Заключительный этап  
Продуктовый сектор  
Междисциплинарные задачи  
9 класс

---

```
print(1)
```

```
n, m = map(int, input().split())
```

```
a = []
```

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

```
    a.append(list(map(int, input().split())))
```

```
prt(n, m, a)
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Задача: Скорость робота**

Ограничение времени 1 секунда

Ограничение памяти 64.0 Мб

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

Перед роботом стоит задача вычисления средней скорости перемещения по маршруту за заданное время  $t$  при равномерном перемещении робота по траектории из четырех точек.

**Формат ввода**

На вход программы поступает 4 строки, каждая из которых состоит из двух вещественных чисел, разделенных пробелом (" ").

Первое число представляет собой координату  $X$  точки маршрута, второе число координату  $Y$  точки маршрута:  $X, Y$ .

$X$  и  $Y$  измеряются в метрах.

$X$  принимает значения из диапазона  $\in [0; 5]$  метров,  $Y$  принимает значения из диапазона  $\in [0; 5]$  метров.

На вход программы в пятой строке поступает время движения по траектории  $t$ , где  $t$  - вещественное число, измеряется в секундах. Ограничения по значению  $t$  отсутствуют.

**Формат вывода**

На выходе программное обеспечение должно выдавать рассчитанные значения длины траектории и значение средней скорости движения робота по траектории.

Значение длины траектории записывается в первую строку. Значение указывается в метрах. Значение необходимо округлить до двух знаков после запятой.

Значение средней скорости движения робота записывается во вторую строку. Значение указывается в м/с. Значение необходимо округлить до двух знаков после запятой.

**Пример 1**

<b>Ввод</b>	<b>Вывод</b>
4.15 5.77	
2.00 3.00	
7.00 6.00	12.94
4.00 8.00	2.06
6.29	

**Пример 2**

<b>Ввод</b>	<b>Вывод</b>
1.0 1.0	4.24
2.0 2.0	2.83
3.0 3.0	

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Ввод**    **Вывод**

4.0 4.0

1.5

### Пример 3

**Ввод**    **Вывод**

2.39 0.29

0.88 2.49

4.37 3.25    8.18

2.81 2.09    0.17

47.0

### Пример решения

```
import math
```

```
def distance(x1, y1, x2, y2):  
    return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
```

```
with open("input.txt", "r") as f:  
    lines = f.readlines()  
    points = [tuple(map(float, line.split())) for line in lines[:4]]  
    time = float(lines[4])
```

```
# Вычисление длины траектории
```

```
trajectory_length = 0  
for i in range(3):  
    x1, y1 = points[i]  
    x2, y2 = points[i + 1]  
    trajectory_length += distance(x1, y1, x2, y2)
```

```
# Вычисление средней скорости
```

```
average_speed = trajectory_length / time
```

```
trajectory_length = round(trajectory_length, 2)
```

```
average_speed = round(average_speed, 2)
```

```
# Запись результатов в файл
```

```
with open("output.txt", "w") as f:  
    f.write(str(trajectory_length))  
    f.write("\n")  
    f.write(str(average_speed))
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Задача: Неизвестная вселенная**

Ограничение времени 1 секунда

Ограничение памяти 64.0 Мб

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

Бороздя неизвестную вселенную, астронавты встретили группу из  $n$  планет радиусом  $R$  км каждая, поверхности которых покрыты жидкостями (плотности жидкостей соответственно равны  $\rho_1, \dots, \rho_n$  кг/м<sup>3</sup>). Высаживаясь на каждую, они проводили следующий эксперимент: на глубину 3 м погружали гирию объемом 0,5 м<sup>3</sup> и плотностью  $\rho_0$  кг/м<sup>3</sup>, затем её начинали вытягивать на поверхность щупом, измеряя проделанную работу. После эксперимента, проделанного на каждой планете, получился набор данных  $A_1, A_2, \dots, A_n$  (Дж).

Определите массу  $M$  ( $10^{24}$  кг) самой тяжёлой из найденных планет по указанным данным.

Использовать гравитационную постоянную  $G = 6.67 \cdot 10^{-11}$  Н·м<sup>2</sup>.

**Формат ввода**

На вход подается  $3 + n \cdot 2$  вещественных числа,  $n, R, \rho_0, \rho_1, A_1, \rho_2, A_2, \dots, \rho_n, A_n$ , каждое число подаётся с новой строки.

**Формат вывода**

Вывести массу  $M$ , самой тяжёлой из найденных планет

**Пример**

Ввод	Вывод
2	
2024	
1900.25	1.001
987.15	
22322	
1000.2	
10950	

**Пример решения**

```
n = int(input())
R = float(input())
rho0 = float(input())
rho = [0] * n
A = [0] * n
M = [0] * n
for i in range(n):
    rho[i] = float(input())
    A[i] = float(input())
    M[i] = round(A[i] * R * 1e6 * R / (3 * 0.5 * (rho0 - rho[i]) * 6.67e13), 3)
print(max(M))
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Задача: Тело бросили под углом к горизонту**

Ограничение времени 1 секунда

Ограничение памяти 64.0 Мб

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

Тело бросили под углом к горизонту.

Однако дует сильный ветер, направленный строго горизонтально в направлении, противоположном движению тела (против оси  $x$ ).

Дано: начальная скорость тела (в м/с), угол наклона начальной скорости к горизонту (в градусах), сила действия ветра (в ньютонах). Масса тела 1 кг.

Ускорение свободного падения считать равным  $g = 10 \text{ м/с}^2$ .

Найдите дальность полёта тела.

Все остальные силы сопротивления в процессе полёта тела не учитываются, учитывается только действие ветра.

**Формат ввода**

На вход подаются данные в виде строки из трех чисел, разделенных пробелом, где первое число – начальная скорость, второе – угол наклона вектора начальной скорости к горизонту в градусах, третье – сила сопротивления ветра в ньютонах.

**Формат вывода**

Необходимо вывести дальность полёта тела в метрах.

Ответ выводить с округлением до двух знаков после запятой.

**Пример**

Ввод	Вывод
10.0 15.0 5.0	4.33

**Примечания**

Если выходным значением является целое число, его необходимо выводить с дробной частью, равной нулю и одним знаком в дробной части.

Разделителем является точка.

В случае получения значений: 1.00 1.00 1.00, 1 1 1, 1.000 1.000 1.000, 1.000000000000001 1.000000000000001, выводить следует: 1.0 1.0 1.0.

**Пример решения**

```
import math
```

```
v0, alfa, F = [float(x) for x in input().split()]
```

**МОСКОВСКАЯ ПРЕПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

```
L = (v0 ** 2 / 10) * math.sin(math.radians(2 * alfa)) - (F * 2 * (v0 ** 2) *  
(math.sin(math.radians(alfa))) ** 2) / 100  
print(round(L, 2))
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Продуктовый сектор**  
**Междисциплинарные задачи**  
**9 класс**

---

**Задача: Часы**

Ограничение времени 1 секунда

Ограничение памяти 256Mb

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

Модест приобрел себе часы, но после запуска их обнаружил, что они неисправны.

Всего на часах  $n$  делений, а большая и маленькая стрелки переключаются на  $x$  и  $y$  делений вперед в час соответственно. Стрелки мгновенно меняют свое положение, т.е. если стрелка находится в положении  $i$ , то через час будет иметь положение  $i + x$ , и стрелки не будут указывать на деления между  $i$  и  $i + x$ .

Изначально стрелки указывали в одно положение. Модесту стало интересно, когда стрелки часов вновь будут указывать в одно положение.

Помогите Модесту разобраться с этими часами.

**Формат ввода**

В первой и единственной строке заданы три целых числа  $n, x, y$  ( $1 \leq n, x, y \leq 1000$ ).

**Формат вывода**

Выведите через сколько часов стрелки вновь будут указывать на одно деление.

**Пример**

Ввод	Вывод
24 7 3	6

**Пример решения**

```
import sys
input = sys.stdin.readline

n, x, y = map(int, input().split())
xx = 0
yy = 0
for i in range(n):
    xx = (xx + x) % n
    yy = (yy + y) % n
    if xx == yy:
        print(i+1)
        exit()
```