

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

### Задача 1 Треугольники

Ограничение времени	3 секунды
Ограничение памяти	256.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У Алисы выходит тройка по математике. Для того, чтобы получить оценку выше тройки, преподаватель дал ей задачу, пояснения к которой, Алиса не смогла найти в учебнике.

Условие к задаче звучит следующим образом:

Дано  $n$  одинаковых равнобедренных прямоугольных треугольников.

Можно ли собрать из данных треугольников **один** квадрат, используя **все** треугольники?

Двигать и переворачивать треугольники можно. Нельзя накладывать друг на друга и нельзя оставлять пустые места внутри квадрата.

Помогите Алисе получить оценку выше тройки.

#### Формат ввода

На вход подается единственное число  $n$  ( $1 \leq n \leq 109$ ) — количество треугольников

#### Формат вывода

Выходными данными является ответ на задачу. YES - если собрать можно и NO если собрать нельзя.

#### Пример 1

Ввод	Вывод
2	YES

#### Пример 2

**Ввод**   **Вывод**

3        NO

МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс

---

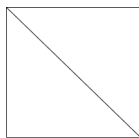
Ввод Вывод

Пример 3

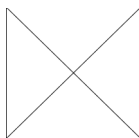
Ввод	Вывод
4	YES

Примечания

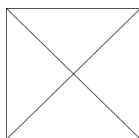
Для  $n = 2$  можно собрать квадрат:



Для  $n = 3$  нельзя собрать квадрат:



Для  $n = 4$  можно собрать квадрат:



МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс

---

Пример решения (GNU c++ 11)

```
#include<iostream>
#include<vector>
#include<cmath>
using namespace std;

int fast_sqrt(int n)
{
    int s = sqrt(n);
    return s * s == n ? s : 1;
}

int main()
{
    int n, a;
    cin >> a;
    if ((a % 2 == 0 && fast_sqrt(a/2) * fast_sqrt(a/2) == a/2)
        || (a % 4 == 0 && fast_sqrt(a/4) * fast_sqrt(a/4) == a/4))
        cout << "YES" << endl;
    else
        cout << "NO" << endl;
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

## Задача 2 Вышивка

Ограничение времени	3 секунды
Ограничение памяти	256.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На праздник 8 Марта девочкам подарили канву для вышивки крестиком. Осталось купить цветные нитки для вышивки. Чтобы рассчитать количество ниток каждого цвета, Маша решила рассчитать количество областей, заполняемых одним цветом. Под областью Маша понимает множество клеток одинакового цвета стыкующихся по **вертикали** или **горизонтали**.

Необходимо посчитать количество областей каждого цвета, кроме белого (цвет самой канвы), и вывести обозначение цветов и количество областей. Гарантируется, что размер схемы не превышает 50x50, количество цветов не превышает 10. Каждый цвет кодируется одним символом (буквой или цифрой). Гарантируется, что белый цвет (цвет канвы) всегда обозначается буквой *w*.

### Формат ввода

В первой строке даются два числа  $N$  и  $M$  – размер вышивки, ( $1 \leq N, M \leq 50$ ). Далее идет  $N$  строк длины  $M$ . В каждой строке через пробел задаются цвета схемы.

### Формат вывода

Программа должна вывести несколько строк (по одной для каждого цвета). Сначала идёт условное обозначение цвета, затем, через пробел, количество областей данного цвета. Цвета необходимо вывести в алфавитном порядке.

Если нет ни одной области ни одного цвета, необходимо вывести строку **NO**.

### Пример 1

Ввод	Вывод
4 5 r r g g r r r w w r w w g w w w g 1 1 w	1 1 g 3 r 2

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

**Пример 2**

Ввод	Вывод
1 1 w	NO

**Примечания**

В примере ниже содержатся:

- одна область с обозначением *1*;
- три области с обозначением *g*;
- две области с обозначением *r*.

R	R	G	G	R
R	R	W	W	R
W	W	G	W	W
W	G	1	1	W

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

**Пример решения (Python 3.10)**

```
def dfs(arr, r, c, n):
    rows, cols = len(arr), len(arr[0])
    if r < 0 or c < 0 or r >= rows or c >= cols or arr[r][c] !=
n:
        return
    arr[r][c] = 'w'
    dfs(arr, r - 1, c, n)
    dfs(arr, r + 1, c, n)
    dfs(arr, r, c - 1, n)
    dfs(arr, r, c + 1, n)

def num_fabric(grid):
    r = len(grid)
    c = len(grid[0])
    if not r:
        return
    count = {}
    for i in range(r):
        for j in range(c):
            if grid[i][j] != 'w':
                count[grid[i][j]] = count.get(grid[i][j], 0) + 1
                dfs(grid, i, j, grid[i][j])
    return count

n, m = map(int, input().split())
canva = []
for i in range(n):
    row = input().split()
    canva.append(row)

fabric = num_fabric(canva)

if not fabric:
    print('NO')

sorted_num_fabric = dict(sorted(fabric.items(), key=lambda x:
x[0].lower()))

for key, value in sorted_num_fabric.items():
    if key == 'w':
        continue
    print("{} {}".format(key, value))
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

### **Задача 3 Ралли**

Ограничение времени	6 секунд
Ограничение памяти	256.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Гоночная трасса разбита на равные части длиной в один километр. Для каждого участка дороги указывается максимальная скорость его прохождения в километрах в час. В начале трассы автомобиль стоит на месте. Перед тем, как начать проезд каждого следующего километрового участка пилот автомобиля может мгновенно ускорить или замедлить автомобиль на определенную величину, определяемую пилотом исходя из скорости прохождения очередного участка. Есть, правда, ограничение на максимальную величину изменения скорости как в сторону увеличения, так и в сторону уменьшения. Таким образом, водитель, зная, какой следует участок перед ним, может рассчитать скорость его прохождения и, либо ускориться, либо замедлиться. Если итоговая скорость даже после максимального торможения слишком велика – проехать этот участок трассы не получится. Хорошая новость – так как Вы знаете сразу всю трассу, и Вы можете заранее правильно рассчитать величины ускорения и торможения перед каждым участком трассы, чтобы итоговое время ее прохождения оказалось минимальным.

#### **Формат ввода**

В первой строке задаются две характеристики автомобиля: максимальное мгновенное ускорение, максимальное мгновенное торможение. Ускорение задается в километрах. Во второй строке передается одно число - длина трассы. Начиная с третьей строки передается трасса. По одной строке на каждый участок дороги. В строке содержится одно число - максимальная допустимая скорость на этом участке в километрах в час. Для справки: длина каждого участка равна одному километру.

#### **Формат вывода**

Набор строк, в каждой строке итоговая скорость прохождения одного участка трассы. То есть, если трасса состоит из 10 участков, на выходе должно быть десять строк с одной цифрой в каждой строке.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс**

---

**Пример**

Ввод	Вывод
50 40	
10	50
100	100
200	90
100	50
50	100
350	140
400	100
100	90
200	50
50	100
500	

**Пример решения (Python 3.10)**

```
import math;

firstLine = input().split();

maxAcceleration = int(firstLine[0]);
maxBrake = int(firstLine[1]);

race = list(map(int, input().split()));

currentSpeed = 0;

speedStack = list();
i = 0;
while (i < len(race)):
    currentAcceleration = 0;
    maxSpeedOnTheSquare = race[i];

    # надо тормозить
    if (currentSpeed > maxSpeedOnTheSquare):
        if (currentSpeed > maxSpeedOnTheSquare - maxBrake):
            # не можем проехать, даже максимально затормозив
            # откатится назад на предыдущие шаги...

            # на сколько нам надо замедлиться
            overSpeed = currentSpeed - maxSpeedOnTheSquare -
maxBrake;

            undoStack = list();
```



МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ. 2021-2022 уч.г.  
Профиль «Информационные технологии»  
Междисциплинарные задачи  
10 класс

---

```
while (True):
    (elSpeed, elAcceleration) = speedStack.pop();

    # высчитаем, насколько мы можем замедлить машину
    decreaseAcceleration = min(overSpeed,
elAcceleration + maxBrake);

    # все в порядке, мы можем замедлиться за один
шаг
    elSpeed -= decreaseAcceleration;
    elAcceleration -= decreaseAcceleration;
    overSpeed -= decreaseAcceleration;
    currentSpeed = elSpeed;

    undoStack.append((elSpeed, elAcceleration));

    if (overSpeed == 0):
        break;

    currentSpeed = speedStack[len(speedStack) - 1][0] if
len(speedStack) > 0 else 0;

    # перенесем откорректированные значения
    while (len(undoStack) > 0):
        (elSpeed, elAcceleration) = undoStack.pop();
        elSpeed = currentSpeed + elAcceleration;
        currentSpeed = elSpeed;
        speedStack.append((elSpeed, elAcceleration));

    currentAcceleration = -max(maxBrake, currentSpeed -
maxSpeedOnTheSquare);

    # надо ускоряться
    elif (currentSpeed < maxSpeedOnTheSquare):
        currentAcceleration = min(maxAcceleration,
maxSpeedOnTheSquare - currentSpeed);

    currentSpeed += currentAcceleration;
    speedStack.append((currentSpeed, currentAcceleration));

    i += 1;

while (len(speedStack) > 0):
    (elSpeed, _) = speedStack.pop();
    print(elSpeed);
```