

## Задача №1

### Некрасивые числа

ограничение по времени на тест : 1 секунда  
ограничение по памяти на тест: 32 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Кубик любит красивые числа. По мнению Кубика, число не считается красивым, если оно делится на квадрат простого числа (Например, число 2527 – некрасивое по мнению Кубика, так оно равно  $7 \cdot 19 \cdot 19$ ). Кубик хочет узнать является ли имеющееся у него число некрасивым. Помогите Кубику решить эту задачу.

#### Входные данные:

Целое число  $a$  ( $1 \leq a \leq 5 \cdot 10^{18}$ ).

#### Выходные данные:

Выведите YES, если число некрасивое, и NO – иначе.

#### Пример

входные данные

1000

выходные данные

YES

#### Решение на Python

```
def simple(n):
    if n==1:
        return False
    for i in range(2, round(n**0.5)+1):
        if n%i==0:
            return False
    return True
n=int(input())
p=False
for i in range(2, round(n**0.5)+1):
    if n%(i*i)==0 and simple(i):
        p=True
        break
if p:
    print("YES")
else:
    print("NO")
```

## Задача 2

### Все по коробкам

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

У Кубика есть  $m$  ящиков и  $n$  коробок. Все стороны каждой коробки равны между собой и характеризуются числом  $r$ , а каждый ящик имеет длину  $a$ , ширину  $b$ , высоту  $c$ . Ящик можно положить в коробку, если он полностью помещается в нем (его стороны строго меньше  $r$ ). Кубик кладет по одному ящику в коробку. Помогите Кубику определить максимальное количество ящиков, которые он сможет положить в коробки.

#### Входные данные:

В первой строке два целых числа  $n, m$  ( $1 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$ ) — количество коробок и ящиков соответственно.

Во второй строке заданы  $n$  целых чисел  $r_1, r_2, \dots, r_n$  характеризующие коробки ( $r_i < 10^5, i = 1..n$ ).

Следующие  $m$  строк содержит 3 целых числа  $a, b, c$  ( $1 \leq a < b \leq 5 \cdot 10^{15}$ ) — размеры ящиков.

#### Выходные данные:

Выведите число — ответ на задачу.

#### Пример

входные данные

```
5 6
1 2 3 4 5
7 7 7
1 2 2
2 3 2
5 5 5
2 3 4
6 6 6
```

выходные данные

```
3
```

#### Примечание

Только ящики 1 2 2, 2 3 2 и 2 3 4 могут быть помещены в коробки.

#### Решение на Python

```
n,m = map(int,input().split())
r=[int(x) for x in input().split()]

d=[]
for i in range (m):
```

```
    d.append(max([int(x) for x in input().split()]))
r=sorted(r)
d=sorted(d)
j=0
answer=0
for i in range(len(d)):
    while (j<len(r)) and (r[j]<=d[i]):
        j+=1
    if j<len(r):
        answer+=1
        j+=1
    else:
        break
print(answer)
```

## Задача 3

### Взять побольше

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Кубик зашел в пиццерию купить пиццу. Каждая пицца имеет диаметр. Все пиццы лежат на длинном прилавке в ряд. Кубик решил взять максимальное количество пицц одного диаметра. В случае если это можно сделать несколькими способами, то он выберет максимальный диаметр. Подскажите Кубику диаметр пиццы, который ему надо выбрать.

#### Входные данные:

В первой строке набора входных данных задано целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество пицц.

Во второй строке заданы  $n$  целых чисел  $d_1, d_2, \dots, d_n$  — диаметры пицц ( $d_i < 10^9, i = 1..n$ ).

#### Выходные данные:

Выведите максимальный диаметр пиццы с максимальным количеством.

#### Пример

входные данные

8

1 2 1 5 1 2 5 2

выходные данные

2

#### Примечание

Максимальное количество пицц одного диаметра равно 3, их диаметры — 1 и 2.

#### Решение на Python

```
n = int(input())
d=[int(x) for x in input().split()]
d=sorted(d)
m=1
answer=d[0]
k=1
for i in range(1,len(d)):
    if d[i-1]==d[i]:
        k+=1
    else:
        k=1
    if k>=m:
        m=k
        answer=d[i]
print(answer)
```

## Задача 4

### Простые пирамидки из кругов

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Кубик складывает пирамидку из кругов. При построении пирамидки Кубик последовательно кладет один круг на другой не меньшего радиуса. Все круги лежат на длинном прилавке в ряд. Кубик последовательно берет круги ряда и строит пирамидку. Если очередной круг не может быть размещен на текущей пирамидке, то Кубик начинает строить новую пирамидку. Также Кубик может в любой момент начать строительство новой пирамидки. Определите сколько пирамидок, в которых количество кругов является максимально возможным простым числом, сможет построить Кубик.

#### Входные данные:

В первой строке задано целое число  $n$  ( $1 \leq n \leq 10^{10}$ ) — количество кругов на прилавке.

Во второй строке каждого набора заданы  $n$  целых чисел  $d_1, d_2, \dots, d_n$  — диаметры кругов ( $d_i < 10^5, i = 1..n$ ).

#### Выходные данные:

Выведите искомое количество, являющегося решением задачи.

#### Пример

входные данные

12

5 6 4 3 2 2 1 8 5 4 2 1

выходные данные

2

#### Примечание

Максимальное количество кругов в одной пирамидке, являющееся простым числом, равно 5. Кубик построит пирамидки: (6, 4, 3, 2, 2), (8, 5, 4, 2, 1)

#### Решение на Python

```
def simple(n):
    if n==1:
        return False
    for i in range(2, round(n**0.5)+1):
        if n%i==0:
            return False
    return True

n = int(input())
d=[int(x) for x in input().split()]
p=[]
p.append(1)
```

```
for i in range(1, len(d)):
    if d[i-1] >= d[i]:
        p[len(p)-1] += 1
    else:
        p.append(1)
m = max(p) + 1
for i in range(max(p), 1, -1):
    if simple(i):
        m = i
        break
answer = 0
for i in p:
    answer += i // m
print(answer)
```

## Задача 5

### Покушать пиццу

ограничение по времени на тест: 45 секунд  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Кубик зашел в пиццерию покушать. Кубик знает, что он наестся только когда скушает ровно 5 пицц. Все пиццы различны. Каждая пицца имеет стоимость. У Кубика есть некая сумма денег. Подскажите Кубику количество вариантов выбора пяти различных пицц на его сумму чтобы наестся.

#### Входные данные:

В первой строке набора входных данных заданы два целых числа  $n$  ( $1 \leq n \leq 10^6$ ) и  $s$  ( $1 \leq s \leq 10^6$ ) — количество пицц и сумма денег у Кубика.

Во второй строке заданы  $n$  целых чисел  $p_1, p_2, \dots, p_n$  — стоимость пицц ( $p_i < 10^9, i = 1..n$ ).

#### Выходные данные:

Выведите искомое количество, являющегося решением задачи.

#### Пример

входные данные

8 9

1 2 5 3 1 2 3 2

выходные данные

7

#### Примечание

Одним из вариантов выбора могут быть пиццы с номерами 1, 5, 6, 7, 8

#### Решение на Python

```
def f(keys,m,max_sum,current_sum, first,count):
    if current_sum<=max_sum and count==0:
        return 1
    if first==len(keys) or current_sum>s or count<=0:
        return 0
    answer = f(keys,m,s,current_sum, first+1,count)
    k=keys[first]
    answer = answer +m[k]*f(keys, m, s, current_sum+k, first + 1, count-1)
    answer = answer +(m[k]*(m[k]-1)//2)*f(keys, m, s, current_sum+2*k, first
+ 1, count-2)
    answer = answer + (m[k] * (m[k] - 1) * (m[k] - 2) // 6) * f(keys, m, s,
current_sum + 3 * k, first + 1,count - 3)
    answer = answer + (m[k] * (m[k] - 1) * (m[k] - 2) * (m[k] - 3)// 24) *
f(keys, m, s, current_sum + 4 * k, first + 1, count - 4)
    answer = answer + (m[k] * (m[k] - 1) * (m[k] - 2) * (m[k] - 3)* (m[k] -
4)// 120) * f(keys, m, s, current_sum + 5 * k, first + 1, count - 5)
    return answer
```

```
n,s = map(int,input().split())
d=[int(x) for x in input().split()]
m={}
for i in d:
    if m.get(i)==None:
        m[i]=1
    else:
        m[i]+=1
k=sorted(m.keys())
print(f(k,m,s,0, 0,5))
```



## Задача 6

### Сходить в гости

ограничение по времени на тест: 10 секунд  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод : стандартный вывод

Кубик решил посетить как можно больше своих друзей, живущих в других городах. Между некоторыми городами ходят автобусы. Кубик хочет узнать, какое максимальное количество друзей он может посетить, передвигаясь на автобусах, если у него имеется определенная сумма денег. Помогите Кубику решить эту задачу, при условии, что Кубик не возвращается в города, которые он посетил.

#### Входные данные:

Первая строка содержит четыре целых числа  $n, s, m, k$  ( $0 \leq n \leq 10^3$ ;  $1 \leq s \leq 3 \cdot 10^6$ ;  $0 \leq m \leq 10^6$ ;  $1 \leq k \leq n$ ) – количество городов, сумма денег, имеющаяся у Кубика, количество маршрутов между городами и номер города в котором живет Кубик.

Затем следуют набор из  $m$  строк, состоящих из трех целых чисел:  $g_1, g_2$  и  $p$  ( $1 \leq g_1 \leq n$ ;  $1 \leq g_2 \leq n$ ;  $g_1 \neq g_2$ ;  $1 \leq p \leq 10^7$ ) где  $g_1$  и  $g_2$  — это номера городов, между которыми ходит автобус и стоимость проезда  $p$ .

#### Выходные данные

Выведите искомое количество, являющегося решением задачи.

#### Примеры

входные данные

6 15 5 2

1 2 3

3 6 2

3 5 6

2 3 4

4 5 2

выходные данные

3

#### Примечание

Кубик сможет посетить друзей в городах с номерами 3, 5, 4.

#### Решение на Python

```
n,s,m,k = map(int, input().split())
neighbours=[[[] for i in range(n)]
prices=[[[] for i in range(n)]
for i in range(m):
    g1,g2,p = map(int, input().split())
    neighbours[g1-1].append(g2-1)
    prices[g1-1].append(p)
    neighbours[g2 - 1].append(g1 - 1)
```

```
prices[g2 - 1].append(p)
def dfs(visited, vertex, rest):
    visited[vertex]=1
    count=0
    for i in range (len(neighbours[vertex])):
        if visited[neighbours[vertex][i]]==0 and rest-prices[vertex][i]>=0:
            count=max(count,1+dfs(visited.copy(),neighbours[vertex][i],rest-
prices[vertex][i]))
    return count
v=[0 for i in range(n)]
print(dfs(v.copy(),k-1,s))
```