

### Задача 1

Коля на летних каникулах занимается ерундой, плохо и неумело прокачивая персонажа в онлайн-игре. Вместо любой предлагаемой активности он ходит по локациям и охотится на монстров, зарабатывая очки опыта. Вдобавок к этому после каждой удачной охоты он пишет, сколько теперь у его персонажа процентов опыта, нужного для повышения уровня. После повышения уровня отсчет начинается заново. Определите по записям, сколько уровней получил Коля, если известно, что никакой монстр не даст ему два уровня сразу. Кроме этого, монстры дают не очень много опыта, поэтому после повышения уровня процент достижения следующего не может превышать предыдущее значение в записях.

#### *Формат ввода*

На вход программе в первой строке подается натуральное число  $N$ , не превышающее **10000** – количество уничтоженных монстров.

Далее в  $N$  строках подается по одному натуральному числу  $v_i$ , не превышающему 99 – процент выполнения задачи в попытке номер  $i$ .

#### *Формат вывода*

Вывести одно целое число – сколько уровней набрал персонаж Коли за летние каникулы.

#### **Пример**

Входные данные	Выходные данные
7 34 56 75 99 34 87 76	2

#### Решение

```
program pzv1;  
  
var  
  s, i, n, pv, v: integer;  
begin  
  readln(n);  
  s:=0;  
  pv:=-1;  
  for i:=1 to n do  
  begin  
    readln(v);  
    if v<pv then
```

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

```
begin  
  s:=s+1;  
end;  
  pv:=v;  
end;  
  writeln(s);  
end.
```

## Задача 2

Группа исследователей управляет зондом. Команды установленному на зонде фотоаппарату подаются вместе с блоком иной информации. Каждый блок представляет собой набор цифр в шестнадцатеричной системе счисления (0123456789ABCDEF). Команда вычисляется на основе последней цифры троичной записи суммы цифр блока:

0: «Снимок».

1: «Снимок со вспышкой»

2: «Очистить память» – очищает память зонда, удаляя все сохраненные фотографии.

Определите максимальное количество фотографий, которые были одновременно удалены из памяти после выполнения  $n$  команд.

*Формат ввода*

В первой строке программе подается на вход число натуральное число  $n$ , не превышающее **1000**.

Далее в каждой из  $n$  строк идет команда – набор цифр в шестнадцатеричной системе счисления (**0123456789ABCDEF**), длина команды не превышает **100** знаков.

*Формат вывода*

Вывести одно число – максимальное количество фотографий, которые были одновременно удалены из памяти после выполнения  $n$  команд.

## Примеры

Входные данные	Выходные данные
4	2
24	
2D	
20	

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

АААВ	
------	--

Решение

```
program z9pzv;

const
  digits = '123456789ABCDEF';

function count(s:string):integer;
var
  i,k:integer;
begin
  k:=0;
  for i:=1 to length(s) do
    begin
      k:=k+pos(copy(s,i,1),digits);
    end;
  count:=k;
end;

var
  n,i,k,max:integer;
  s:string;
begin
  readln(n);
  max:=0;
  k:=0;
  for i:=1 to n do
    begin
      readln(s);
      if (count(s) mod 3) <> 2 then
        begin
          k:=k+1;
        end
      else
        begin
          if k > max then
            max:=k;
          k:=0;
        end;
      end;
    writeln(max);
  end.
```

### Задача 3

В ходе игры «Зарница» Витя и Паша пересылают друг другу важные сообщения. Но для того, чтобы противник не смог их понять, сообщения кодируются. Для кодирования информации ребята используют латинский

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

алфавит из 26 букв, все буквы заглавные. Слова кодируются следующим образом. Каждая буква в слове заменяется ее порядковым номером в алфавите, записанном в системе счисления с основанием  $Sys$  ( $2 \leq Sys \leq 36$ ). Все полученные числа записываются подряд без пробелов. Если числа (порядковые номера букв) в заданной системе счисления могут иметь разную длину, то более короткие числа дополняются слева нулями до требуемой длины. Например, в десятичной системе счисления порядковый номер буквы **A** будет равен **1**, а буквы **Z** – **26**. Соответственно, при шифровании, к единице слева будет дописан ноль. То есть код буквы **A** будет **01**, а код буквы **Z** – **26**. Для усложнения возможной расшифровки сообщения противником, при кодировании разных слов, могут использоваться различные системы счисления. Основание использованной системы счисления, выраженное двухзначным десятичным числом дописывается справа к коду всего слова.

Например, слово **AZ**, при использовании десятичной системы счисления, будет закодировано как **012610**, а при использовании троичной системы счисления будет закодировано как **00122203**.

Напишите программу, которая будет расшифровывать закодированные сообщения.

На вход программе подается одно закодированное сообщение. Длина сообщения не более 100 символов. Программа должна вывести исходное слово.

### Примеры

Входные данные	Выходные данные
02201211011012003	HELLO
122130	ABBA

### Решение

```
alf = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  
cifr = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'  
  
s = input()  
sys = int(s[-2:])  
if sys >= 26:  
    lenb = 1  
elif sys >= 6:
```

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

```
lenb = 2
elif sys >= 3:
    lenb = 3
else:
    lenb = 5
s = s[:-2]
res = ''
for i in range(0, len(s), lenb):
    ns = s[i:i + lenb]
    p = int(ns, sys)
    res += alf[p - 1]

print(res)
```

#### Задача 4

На курсы по изучению Python записалось  $N$  ( $0 < N \leq 100$ ) человек. Необходимо составить расписание занятий. Для этого необходимо определить дни, когда на занятиях сможет присутствовать максимальное количество учащихся. Каждый учащийся заполнил специальную анкету, где указал дни, в которые он может посещать курсы. Пожелания были двух типов.

- 1) Дни недели. Обозначение: строка начинается с буквы “w”; далее идет цепочка цифр без пробелов: 1 – понедельник, 2 – вторник, 3 – среда, 4 – четверг, 5 – пятница, 6 – суббота, 7 – воскресенье. Например, в строке “w134” – закодировано, что учащийся может проходить обучение по понедельникам (1), средам (3) и четвергам (4).
- 2) Чётные/нечётные даты. Обозначение: строка начинается с буквы “d”; далее идет одна цифра: 1 – нечётные даты, 0 – чётные даты. Например, в строке “d1” – закодировано, что учащийся может проходить обучение по нечетным датам (1, 3, 5, 7, ..., 29, 31).

Каждый учащийся высказал пожелание только одного типа.

Необходимо определить, сколько дней в период обучения таковы, что на занятиях может присутствовать максимальное количество учащихся.

Гарантируется, что дата начала меньше даты окончания обучения, что обучение начинается не ранее 1 января 2023 года. Период не превышает двух календарных лет. 01.01.2023 – воскресенье (остальные дни недели должна определять программа).

#### Входные данные:

Построчно.

Дата начала интервала обучения в формате dd.mm.yyyy

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

Дата окончания интервала обучения (включая этот день) в формате dd.mm.yyyy

N – количество учащихся.

В каждой последующей строчке закодированы пожелания учащихся (формат смотри выше).

**Выходные данные:**

Два числа через пробел.

Количество дней с максимальным количеством учащихся, планирующих посетить занятия.

Планируемое максимальное количество учащихся на занятии.

Входные данные	Вывод	Примечание
02.01.2023 16.01.2023 5 w134 d0 w123 w15 d0	2 5	Все пятеро учащихся смогут присутствовать на занятиях 2 и 16 января 2023 года (два дня).
02.01.2023 16.01.2023 5 w134 d0 w123 w15 d1	3 4	На занятиях максимально может присутствовать только четверо из пятерых учащихся. Даты, когда это возможно: 2, 9 и 16 января 2023 года (три дня).
10.01.2023 13.01.2023 5 w135 d1 w13 w15 d1	2 4	А 11 и 13 января будет максимальное количество человек - 4 (два дня).

**Решение**

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

**(без библиотек)**

```
def v(y):      # проверка года на високосность
    if y % 400 == 0: return True
    if y % 100 == 0: return False
    if y % 4 == 0: return True
    return False

def f(m, y):   # количество дней в месяце m данного года y
    if m == 2 and v(y):
        return 29
    days=[0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    return days[m]

def next(d,m,y): # следующая дата в календаре
    d += 1
    if d > f(m,y):
        d = 1
        m += 1
        if m > 12:
            m = 1
            y += 1
    return d, m, y

def weekday(d,m,y): # день недели
    ans = d
    mm, yy = 1, 2023
    while not(mm == m and yy == y):
        ans += f(mm, yy)
        mm += 1
        if mm > 12:
            mm = 1
            yy += 1
    w = (ans - 1) % 7
    return w

d1, m1, y1 = map(int, input().split('.'))
d2, m2, y2 = map(int, input().split('.'))
cnt = [[0] * 7 for _ in range(2)]
w = weekday(d1, m1, y1)
cnt[d1 % 2][w] += 1

while not(d1 == d2 and m1 == m2 and y1 == y2):
    d1, m1, y1 = next(d1, m1, y1)
    w = (w + 1) % 7
    cnt[d1 % 2][w] += 1

x = [[0] * 7 for _ in range(2)]
n = int(input())
for _ in range(n):
    s = input()
    if s[0] == 'w':
```

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

```
for c in s[1:]:
    x[0][int(c) % 7] += 1
    x[1][int(c) % 7] += 1
else:
    for j in range(7):
        x[int(s[1])][j] += 1

lx = [x[0][i] for i in range(7) if x[0][i] != 0 and cnt[0][i] != 0]
rx = [x[1][i] for i in range(7) if x[1][i] != 0 and cnt[1][i] != 0]

mx = max(lx + rx)
ans = 0
for i in range(2):
    for j in range(7):
        if x[i][j] == mx:
            ans += cnt[i][j]
print(ans, mx)
```

## Решение

### (с библиотекой)

```
from datetime import date, timedelta

d1, m1, y1 = map(int, input().split('.'))
d2, m2, y2 = map(int, input().split('.'))
cnt = [[0] * 7 for _ in range(2)]
a = date(y1, m1, d1)
b = date(y2, m2, d2)
one = timedelta(days=1)

while a <= b:
    tt = a.timetuple()
    y1, m1, d1 = tt[0], tt[1], tt[2]
    w = (a.weekday() + 1) % 7
    cnt[d1 % 2][w] += 1
    a += one

x = [[0] * 7 for _ in range(2)]
n = int(input())
for _ in range(n):
    s = input()
    if s[0] == 'w':
        for c in s[1:]:
            x[0][int(c) % 7] += 1
            x[1][int(c) % 7] += 1
    else:
        for j in range(7):
            x[int(s[1])][j] += 1
```

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

```
lx = [x[0][i] for i in range(7) if x[0][i] != 0 and cnt[0][i] != 0]
rx = [x[1][i] for i in range(7) if x[1][i] != 0 and cnt[1][i] != 0]

mx = max(lx + rx)
ans = 0
for i in range(2):
    for j in range(7):
        if x[i][j] == mx:
            ans += cnt[i][j]
print(ans, mx)
```

### Задача 5

Город имеет форму прямоугольника с вершинами в точках  $(-W, -H)$ ,  $(-W, H)$ ,  $(W, H)$ ,  $(W, -H)$ .

Плоскость разбита на кварталы. Квартал — это единичная клетка, вершины которой имеют целочисленные координаты. Назовем квартал городским, если *все* вершины квартала находятся внутри города (считается, что точка на границе принадлежит городу). Всего в городе будет  $4 \cdot W \cdot H$  кварталов.

Дорожная сеть состоит из  $N$  дорог (часть дорог или все проходят через город).

Дорога — это прямая линия, не параллельная осям координат.

Дорога задается двумя различными точками на ней (точки могут находиться вне города).

Для каждого квартала определим "*значимость*". *Значимость* квартала равна количеству дорог, проходящих через этот квартал. Считается, что дорога проходит через квартал, если имеет с кварталом не менее двух общих точек.

Найдите значение "*значимости*" для каждого квартала. Для каждой полученной "*значимости*" определите количество кварталов, имеющих эту *значимость*.

#### Входные данные

В первой строке заданы значения  $W, H, N$  ( $9 < W, H < 201$ ,  $0 < N < 1001$ )

В следующих  $N$  строках задано по четыре числа (координаты двух точек прямой, определяющих дорогу).

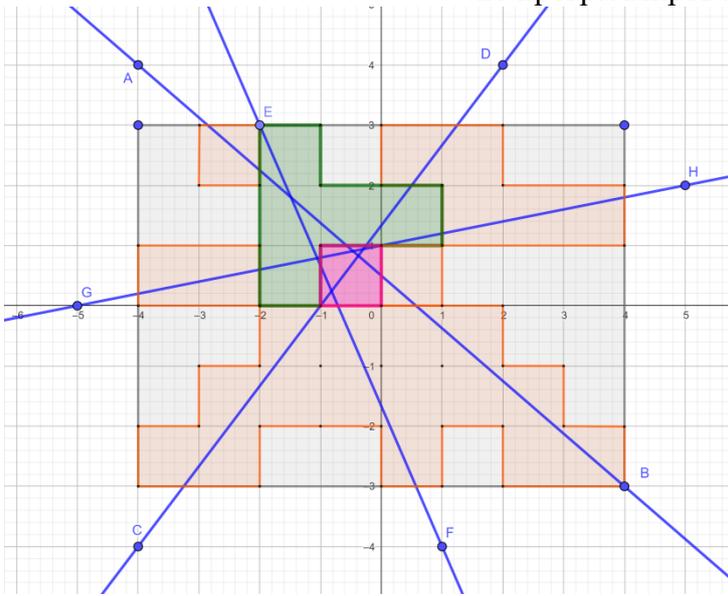
#### Выходные данные

В первой строке выведите число  $K$  - количество различных ненулевых значений "*значимости*".

В следующих  $K$  строках выведите по два числа - значение "*значимости*" и количество кварталов, имеющих такое значение "*значимости*".

#### Пример

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию



Входные данные	Выходные данные	Пояснение
<pre> 4 3 4 -4 4 4 -3 -4 -4 2 4 -2 3 1 -4 -5 0 5 2                     </pre>	<pre> 3 1 24 2 5 4 1                     </pre>	<p>Город расположен в прямоугольнике со сторонами 8 и 6 клеток (всего 48 кварталов)</p> <p>Через город проходят 4 дороги AB, CD, EF, GH</p> <p>Значимость 1 будет у 24 кварталов (коричневый цвет на рисунке)</p> <p>Значимость 2 будет у 5 кварталов (зеленый цвет на рисунке)</p> <p>Значимость 4 будет у 1 кварталов (красный цвет на рисунке)</p> <p>18 кварталов будут иметь значимость равную 0 (на печать не выводиться)</p>

## Решение

```

def nod(a,b): # Нахождение НОД для сокращения коэф. уравнения прямой
    while b>0 :
        a,b=b,a %b
    return a

def urav (ab) : # уравнение прямой через две точки вида x*kx+y*ky+kc=0
    ax,ay=ab[0][0],ab[0][1] # координаты 1-й точки
    bx,by=ab[1][0],ab[1][1] # координаты 2-й точки
    vx,vy=bx-ax,by-ay # координаты вектора прямой
    d=nod(abs(vx),abs(vy)) # общий делитель
    vx,vy=vx//d,vy//d # сокращение координат вектора
    if vy<0 : vx,vy=-vx,-vy # коэф. kx делаем положительным
                    
```

**Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию**

```
return vy, -vx, -ax*vy+ay*vх # коэф. уравнения прямой

W, H, N=map(int, input().split()) # размер города, кол-во дорог
rez=[0]*(N+1) # массив кварталов с заданной значимостью
W*=2; H*=2 #длина и ширина города
YX=[] # массив кварталов (квартал задается левым нижним углом)
for _ in range (H): # заполнение YX начальными нулями
    A=[0]*W
    YX.append(A)

# Основная обработка. Цикл по дорогам. Для каждой дороги определяем
# пересекаемые кварталы (почти линейно от W+H) и учитываем в результате
# !!! После каждой дороги доступен результат
# Существенно используется, что дороги не параллельны сторонам города
# Для удобства делается переход к НОВЫМ координатам, в которых
# Юго-Запад города имеет координаты (0,0) вместо (-W,-H)
# ЗНАЧИМОСТЬ - кол-во дорог, проходящих через квартал

for i in range(N): # Основной цикл по дорогам
    ax, ay, bx, by=map(int, input().split()) # ввод координат 2 точек на дороге
    if ax>bx : ax, ay, bx, by=bx, by, ax, ay # прическа ax<ay
    A, B=(ax+W//2, ay+H//2), (bx+W//2, by+H//2) # к новым координатам
    kx, ky, kc=urav((A, B)) # целые коэф. уравнения прямой АВ
    if ky<0 : # y=kx+c дорога с ЮЗ на СВ
        x0=max(0, -kc/kx) # start по X
        x1=min(W, (-H*ky-kc)/kx) # finish по X
    if ky>0 : # y=-kx+c дорога с СЗ на ЮВ
        x0=max(0, (-H*ky-kc)/kx) # start по X
        x1=min(W, -kc/kx) # finish по X
    x0, x1=int(x0), min(int(x1)+1, W) # целые пределы интервала по X
    for x in range (x0, x1): # цикл по X для маркировки кварталов
        y0=(-x*kx-kc)/ky # точка пересечения с прямой y=x
        y1=(-x*kx-kx-kc)/ky # точка пересечения с прямой y=x+1(следующий x)
        if ky<0 : #дорога в положительном направлении y0<=y1 (ЮЗ-СВ)
            y0=max(-1, int(y0)) # к целому
            y1=min(H, int(y1-10**(-9))) #со смещением (для целого знач. y1)
        if ky>0 : #дорога в отрицательном направлении y0>=y1
            y0=min(H, int(y0-10**(-9))) #со смещением (для целого знач. y0)
            y1=max(-1, int(y1)) # к целому
        y0, y1=min(y0, y1), max(y0, y1) # интервал по y для заполнения
        for y in range (max(0, y0), min(H, y1+1)): # заполнение результатов
            r=YX[y][x] # старое значение ЗНАЧИМОСТИ
            rez[r]-=1 # убираем из итогового массива
            rez[r+1]+=1 # добавляем новое
            YX[y][x]+=1 # изменяем для квартала

# Вычисление итогов, вывод результатов
rk=0 # количество различных значений ЗНАЧИМОСТИ
for r in rez : # подсчет rk
    if r>0 : rk+=1
print(rk) # вывод на печать
for i in range(N+1): # вывод ненулевых значений ЗНАЧИМОСТИ КВАРТАЛОВ
    if rez[i]>0 :
        print(i, rez[i])
```

### Задача 6

Профессор Чадов и аспирант Шлёпов оптимизируют производство октогена. Одним из важных компонентов для создания этой мощной взрывчатки является азотная кислота. Чтобы как можно меньше таскать сосуды с кислотой, лаборанты попросили аспиранта Шлёпова написать программу, которая будет рассчитывать, какие емкости надо принести со склада в лабораторию, чтобы выполнялись несколько условий:

- 1) Объем азотной кислоты должен быть не меньше требуемого для работы;
- 2) Объем азотной кислоты в лаборатории должен быть минимально возможным;
- 3) При прочих равных следует предпочесть переноску меньшего количества емкостей;

Напишите программу, которая поможет лаборантам.

#### *Формат ввода*

В первой строке программы вводится натуральное число  $N$  ( $N \leq 20$ ) – количество емкостей с кислотой. Во второй строке указывается натуральное число  $V$  ( $0 \leq V \leq 200$  л) – ограничение по объему. Далее в  $N$  строчках вводится по одному натуральному числу  $v_i$  ( $v_i \leq 20$  л) – объем емкости под номером  $i$ .

#### *Формат вывода*

Вывести в одной строке через пробел в порядке возрастания объемы емкостей, которые надо отнести в лабораторию, уложившись в заданные условия. Если это невозможно, вывести 0.

#### Пример

Входные данные	Выходные данные
4 20 19 9 5 6	5 6 9

#### Решение

```
from itertools import combinations
```

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

```
v = int(input())
```

Заключительный этап Олимпиады школьников «Шаг в будущее»  
по программированию

```
data = []

for i in range(n):
    data.append(int(input()))

answer = data.copy()

if sum(data)>=v:
    for i in range(1,n+1):
        var = combinations(data,i)
        for x in var:
            if (sum(x)>=v) and (sum(x)<=sum(answer)):
                answer = list(x)

else:
    answer = [0]

answer.sort()
print(*answer)
```

**КРИТЕРИИ ОЦЕНИВАНИЯ ОЛИМПИАДНЫХ ЗАДАНИЙ ЗАКЛЮЧИТЕЛЬНОГО  
ЭТАПА ОЛИМПИАДЫ ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»**

Максимальная сумма баллов за выполненные задания варианта – 100.

8 - 11 классы

Распределение баллов по заданиям:

Номер задачи	1	2	3	4	5	6 Ситуационная задача
Баллы	8	12	16	20	20	24

Балл за каждую задачу проставляется в соответствии с долей выполнения участником задания. Шаг оценивания каждого задания - 1 балл. Доля выполнения вычисляется автоматической системой тестирования на основе комплекта установленных для задания тестов.