

**Заключительный этап академического соревнования Олимпиады школьников  
«Шаг в будущее» по программированию «Профессор Лебедев»  
(общеобразовательный предмет информатика), весна 2021 год**

**10 класс**

**Вариант 1**

**Задача 1**

Интернет-тролль Ибрагим просыпается в 8 часов ровно и сразу же начинает без остановки строчить сообщения в соцсети, провоцируя пользователей на эмоциональную переписку с переходом на личности. На написание и отправку каждого сообщения уходит 30 секунд.

Юзер Юся просыпается в  $h$  часов  $m$  минут, сразу же залезает в соцсеть и начинает потреблять сегодняшние сообщения, тратя по  $n$  секунд на каждое. И, если сообщение тролля уже написано, потребляет именно его.

Устойчивость юзера Юси к провокациям составляет  $k$  сообщений тролля, после прочтения которых юзер ввязывается в переписку тем, что тут же после прочтения  $k$ -го сообщения тролля пишет первое сообщение сам.

Временем начала переписки считается время начала написания юзером первого сообщения.

Сколько времени будет на часах, когда интернет-тролль Ибрагим достигнет своей цели?

На вход программе в одной строке через пробел подаются 4 целых числа – данные юзера:  $h$ ,  $m$ ,  $n$  и  $k$ . Время, когда просыпаются тролль и юзер, указано в одних сутках.  $n$ ,  $k > 0$ . Исходные данные заданы так, что время начала переписки гарантированно будет в тех же сутках.

В одной строке выходных данных через пробел программа должна вывести искомое время: час, минуту и секунду, которые будут на часах в момент начала переписки.

**Пример**

Входные данные	Выходные данные
8 0 30 4	8 2 30

Комментарий к примеру

Время начал написания сообщений: 8:00:00, 8:00:30, 8:01:00, 8:01:30.

Время, когда юзер будет начинать читать сообщения: 8:00:30, 8:01:00, 8:01:30, 8:02:00, после чего в 8:02:30 начнёт писать своё.

**Тесты**

Входные данные	Выходные данные
8 0 30 4	8 2 30
8 5 20 4	8 6 20
9 0 40 99	10 6 0
9 0 40 98	10 5 20
13 34 10 1000	16 20 40
13 35 10 1000	16 21 40
7 0 30 4	8 2 30
7 0 60 4	8 5 0

7 0 10 6	8 3 10
7 0 25 7	8 4 10

### Решение

```

hi, mi, ni = 8, 0, 30
hu, mu, nu, k = map(int, input().split())
i = (hi * 60 + mi) * 60
u = (hu * 60 + mu) * 60
#          время          окончания          написания          k          сообщения
ik          =          i          +          ni          *          k
# время          окончания          написания          троллем          первого          сообщения
i1 = i + ni
# если юзер проснулся не раньше возможности прочитать первое сообщение тролля
if          u          >=          i1:
    # временем          начала          чтения          является          время,          когда          юзер          проснулся
    start          =          u
else:
    # временем          начала          чтения          является          время          окончания          чтения          после          написания
    start = (i1 - u + nu - 1) // nu * nu + u
# время, когда юзер мог бы прочитать сообщения, если бы они были уже написаны
finish = start + k * nu
# если чтение было подряд (время окончания написания k сообщений не позже начала чтения
# последнего сообщения), то
if          ik          <=          finish          -          nu:
    res          =          finish
else:
    # временем          начала          чтения          последнего          сообщения
    start = (ik - u + nu - 1) // nu * nu + u
    res = start + nu
s = res % 60; res //= 60 # переходим в минуты
m = res % 60; res //= 60 # переходим в часы
h = res % 24; res //= 24 # переходим в дни
print(h, m, s)

```

### Задача 2

В некоторой стране Букварии люди вели календарь особенным образом. Для записи номера года они использовали строчные символы английского алфавита: a – 1, b – 5, c – 10, d – 50, e – 100, f – 500, g – 1000. При этом календарь отсчитывал года от момента образования этой страны.

Номера записывались слева направо по следующим правилам: каждый меньший знак, поставленный слева от большего, вычитается из него, а каждый меньший знак, поставленный справа от большего, прибавляется к нему.

Однажды к ним в страну приехал Незнайка. На одном из плакатов был написан текущий год. Незнайка любил историю и ему было очень интересно, когда была образована эта страна, но понимал только десятичные цифры. Помогите Незнайке определить, когда же появилась Буквария. Напишите программу, которая будет переводить номер года страны Букварии в десятичное число.

### ВХОДНЫЕ ДАННЫЕ

На вход подается последовательность символов (a-g).

### ВЫХОДНЫЕ ДАННЫЕ

Число, полученное в результате преобразований.

### Пример

Входные данные	Выходные данные
cab	14

### Тесты

ВХОДНЫЕ ДАННЫЕ	ВЫХОДНЫЕ ДАННЫЕ
сса	21
ab	4
dac	59
ggfedcba	2666
egcccbab	939
ed	150
a	1
b	5
c	10
d	50
e	100
f	500
g	1000

### Решение (Python3)

```
def solution(roman):
    nums = {'a': 1, 'b': 5, 'c': 10, 'd': 50, 'e': 100, 'f': 500, 'g': 1000}
    lst = [nums[i] for i in roman]
    i = 0
    s = 0
    l = len(lst)
    while i < l:
        if i < l - 1 and lst[i] < lst[i + 1]:
            s += lst[i + 1] - lst[i]
            i += 1
        else:
            s += lst[i]
            i += 1
    return s
```

### Задача 3

Автомат получает на вход натуральное число  $X$  (не превышающее  $10^6$ ). По этому числу строится новое *шестнадцатеричное* число  $Y$ :

- первая цифра нового числа – это крайняя правая цифра восьмеричного представления  $X$ ;
- вторая цифра числа  $Y$  – это крайняя правая цифра четверичного представления  $X$ ;
- третья цифра числа  $Y$  – это крайняя правая цифра двоичного представления  $X$ .

Далее цифры выстраиваются по невозрастанию, а затем полученное шестнадцатеричное число переводится в десятичное - результат работы автомата число  $Y$ .

Например,  $X = 190_{10} = 276_8 = 2332_4 = 10111110_2$ . Тогда новое число  $Y = 620_{16} = 1568_{10}$ .

Напишите программу, которая будет искать наименьшее из чисел заданного интервала  $[A, B]$ , на котором автоматом будет получено наибольшее число, а также сколько раз на данном интервале будет выдано наибольшее значение.

На вход программы подаётся два целых числа  $A$  и  $B$  ( $1 \leq A \leq B \leq 1\,000\,000$ ), записанных через пробел.

Программа должна вывести в одной строке через пробел два целых числа - вычисленные значения.

Входные данные	Вывод	Примечание
188 190	190 1	188 -> 1024 189 -> 1297 190 -> 1568 наибольшее

### Тесты

Ввод	Вывод
1 10	7 1
512 512	512 1
1968 1990	1975 2
1 1000000	7 125000
999900 1000000	999903 13

### Решение

```
program automatic1;
```

```
function automatic(x: longint): longint;
```

```
var digit1, digit2, digit3: integer;
```

```
begin
```

```
  digit1 := x mod 8;
```

```
  digit2 := x mod 4;
```

```
  digit3 := x mod 2;
```

```
  automatic := digit1 * 256 + digit2 * 16 + digit3
```

```
end;
```

```
var a, b, n, max, y, k, xmax: longint;
```

```
begin
```

```
  readln(a, b);
```

```
  max := -1;
```

```
  k := 0;
```

```
  for n := a to b do
```

```
  begin
```

```
    y := automatic(n);
```

```
    if y > max then
```

```
    begin
```

```
      max := y;
```

```
      xmax := n;
```

```
      k := 1
```

```
    end
```

```
    else
```

```
      if y = max then
```

```
        k := 1 + k;
```

```
    end;
```

```
  write(xmax, ' ', k)
```

```
end.
```

#### Задача 4

Робота можно разместить на разных гусеничных шасси. Зная массу робота, массу каждого типа шасси и площадь поверхности его гусениц, найдите шасси, чье давление на грунт после установки робота ближе всего к среднему арифметическому давлений представленных шасси после установки на них робота. Все шасси отличаются давлением, которое они оказывают.

*Формат ввода*

В первой строке вводится натуральное число  $N$  ( $1 \leq N \leq 15$ ) – количество шасси. Во второй строке вводится вещественное число  $M$  – масса робота. Далее в  $N$  строчках вводится по два вещественных числа  $m_i$  и  $s_i$  – масса и площадь гусениц шасси под номером  $i$ . Все величины указываются в системе СИ, имеют физический смысл и не превосходят 10000.

*Формат вывода*

Вывести через пробел одно целое число – номер шасси, чье давление на поверхность ближе всего к среднему арифметическому давлений представленных шасси после установки на них робота.

#### Пример

Входные данные	Выходные данные
4	4
5	
2 3	
3 3	
6 3	
4 3	

#### Тесты

Входные данные	Выходные данные
4	4
5	
2 3	
3 3	
6 3	
4 3	
4	4
1	
1 2	
8 3	
3 1.7	
3 2.1	
1	1
10	
1 1	

4 1 1 1 3 1.9 3 2.3 8 3	2
5 5 1.6 1 2.3 3 2.5 2 3.6 4 5 9	3

### Решение

```
program mzv1;
```

```
const
```

```
  K=15;
```

```
type
```

```
  tmas = array [1..K] of real;
```

```
var
```

```
  mass,squ:tmas;
```

```
  sr,m:real;
```

```
  n,i,mi:integer;
```

```
begin
```

```
  readln(n);
```

```
  readln(m);
```

```
  sr:=0;
```

```
  for i:=1 to n do
```

```
    begin
```

```
      readln(mass[i],squ[i]);
```

```
      sr:=sr+(mass[i]+m)/squ[i];
```

```
    end;
```

```
  sr:=sr/n;
```

```
  mi:=1;
```

```
  for i:=1 to n do
```

```
    if abs((mass[i]+m)/squ[i]-sr)<abs((mass[mi]+m)/squ[mi]-sr) then
```

```
      begin
```

```
        mi:=i;
```

```
      end;
```

```
  writeln(mi);
```

```
end.
```

### Задача 5

Трое разведчиков, пронумерованных как **0**, **1** и **2**, передают зашифрованные сообщения в штаб. Сообщения состоят только из строчных латинских букв и цифр. Номер разведчика, отправившего сообщение, вычисляется как остаток от деления суммы цифр в сообщении на 3. Определить, какой разведчик отправил больше всего сообщений. Если в сообщении нет цифр, оно считается шумом.

*Формат ввода*

В первой строке вводится сначала натуральное число  $n$  ( $n \leq 1000$ ) – количество сообщений, затем в  $n$  следующих строках записано по одному сообщению.

*Формат вывода*

Вывести одно целое число – номер разведчика, который передал больше всего сообщений в штаб. Если два разведчика передали максимальное количество сообщений, вывести больший номер.

### Примеры

Входные данные	Выходные данные
3 a3b a23c a0c	0

### Тесты

Входные данные	Выходные данные
3 a3b a23c a0c	0
5 ab0 acb dfr abd1 fda4	1
4 ab2 ac2 ba1 ab4	2
6 afd ads fd asdr dfg	2



dfr	
3 ab1 ac1 fd1	1

### Решение

```
program szv1;
```

```
const
```

```
digits = '1234567890';
```

```
function good(s:string):boolean;
```

```
var
```

```
i:integer;
```

```
flag:boolean;
```

```
begin
```

```
flag:=false;
```

```
i:=1;
```

```
while (i<=10) and not flag do
```

```
begin
```

```
flag:=pos(copy(digits,i,1),s)>0;
```

```
i:=i+1;
```

```
end;
```

```
good:=flag;
```

```
end;
```

```
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) mod 10);
```

```
end;
```

```
count:=k mod 3;
```

```
end;
```

```
var
```

```
n,k,i,a,b,c:integer;
```

```
s:string;
```

```
begin
```

```
readln(n);
```

```
a:=0;
```

```
b:=0;
```

```
c:=0;
```

```
for i:=1 to n do
```

```
begin
```

```
readln(s);
```

```

if good(s) then
  begin
    k:=count(s);
    if k = 0 then
      a:=a+1;
    if k = 1 then
      b:=b+1;
    if k = 2 then
      c:=c+1;
    end;
  end;
if (a>b) and (a>c) then
  writeln(0);
if (b>=a) and (b>c) then
  writeln(1);
if (c>=a) and (c>=b) then
  writeln(2);
end.

```

### Задача 6 Ситуационная задача

Легкий прочный стержень образует примитивные рычажные весы. Известно, во сколько раз один рычаг длиннее другого. На концы стержня можно подвешивать по несколько гирь. Зная массы гирь, определите, какую пару надо подвесить на один из концов стержня, чтобы они оказались как можно ближе к положению равновесия с грузом заранее известной массы **M**.

#### Формат ввода

В первой строке программы вводится вещественное положительное число **D** – во сколько раз первый рычаг длиннее второго. Во второй строке программы вводится положительное вещественное число **M** – масса груза. В третьей строке вводится натуральное число **N** ( $2 \leq N \leq 15$ ) – количество гирь. Далее в **N** строчках вводится по одному положительному вещественному числу **m<sub>i</sub>** – масса гири под номером **i**. Масса указывается в граммах. Масса гири не превышает 100 г, масса груза не превышает 1000 г.

#### Формат вывода

Вывести через пробел два целых числа – номера пары гирь, которые будут ближе всего к положению равновесия с грузом массы **M**. Требуется вывести номера грузов в порядке возрастания. На какой стержень надо подвесить гири, указывать не надо. Если таких пар несколько, вывести пару с меньшим минимальным номером и как можно меньшей суммой номеров.

### Пример

Входные данные	Выходные данные
2.5 100.0 4 10.8 14.9 25.1 19.2	2 3

### Тесты

Входные данные	Выходные данные
2.5 100.0 4 10.8 14.9 25.1 19.2	2 3
2 100 4 10 20 30 60	2 3
5 10 2 1 1	1 2
1 100 5 10 80 90 20 50	1 3
0.1 900 7 10 20 60 150 30	3 5

10	
10	

**Решение**

**program** s10zv1;

**const**

K=15;

**type**

tmas = **array** [1..K] **of** real;

**var**

mas:tmas;

d,m,min:real;

n,i,j,mi,mj:integer;

**begin**

readln(d);

readln(m);

readln(n);

mi:=K+1;

mj:=K+1;

**for** i:=1 **to** n **do**

**begin**

readln(mas[i]);

min:=min+mas[i];

**end;**

min:=min\*d;

**for** i:=1 **to** n-1 **do**

**for** j:=i+1 **to** n **do**

**begin**

**if** (abs((mas[i]+mas[j])\*d-m) < min) **then**

**begin**

mi:=i;

mj:=j;

min:=abs((mas[i]+mas[j])\*d-m);

**end;**

**if** (abs((mas[i]+mas[j])-m\*d) < min) **then**

**begin**

mi:=i;

mj:=j;

min:=abs((mas[i]+mas[j])-m\*d);

**end;**

**end;**

writeln(mi,' ',mj)

**end.**