

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

**Вариант 1**

**Задача 1**

Интернет-троль 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-троль 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**

$h_i, m_i, n_i = 8, 0, 30$

$h_u, m_u, n_u, k = \text{map}(\text{int}, \text{input}().\text{split}())$

```

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  $X$ ;
- последняя цифра числа  $Y$  – это крайняя правая цифра двоичного представления  $X$ .

Результат работы автомата – десятичное представление построенного числа  $Y$ .

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

Напишите программу, которая будет искать наибольшее число, которое может быть получено автоматом из чисел заданного интервала  $[A, B]$ .

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 одно целое число - вычисленное значение.

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

### Тесты

Ввод	Вывод
1 10	1841
512 512	0
1968 1971	817
1 1000000	1841
999990 1000000	1841

### 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: longint;
```

```
begin
```

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

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

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

```
    begin
```

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

```
      writeln(y);
```

```
      if y > max then
```

```
        max := y
```

```
    end;
```

```
  write(max)
```

```
end.
```

#### Задача 4

Для сборки конструкции нужны два поплавка. Зная массу каждого из поплавков, имеющихся в наличии, найти номера самого легкого и самого тяжелого поплавка.

*Формат ввода*

В первой строке программы вводится натуральное число  $N$  – количество поплавков.  $2 \leq N \leq 300$ . Далее в  $N$  строчках вводится по одному натуральному числу  $m_i$  ( $m_i \leq 1000$  г) – масса поплавок под номером  $i$ .

*Формат вывода*

Вывести через пробел два целых числа – номера самого легкого и самого тяжелого поплавка. Сначала вывести наименьший из номеров. Если у двух поплавков одинаковая масса, вывести меньший номер.

#### Пример

Входные данные	Выходные данные
5 4 3 6 5 8	2 5

#### Тесты

Входные данные	Выходные данные
5 4 3 6 5 8	2 5
4 1 2 3 4	1 4
2 2 2	1 1
8 4 3 5 6 7 7 1 4	5 7

5 4 2 3 5 1	1 5
----------------------------	-----

### Решение

**program** pzv1;

**var**

i,n,num,pnum:integer;

x,min,pmin:real;

**begin**

readln(n);

min:=1001;

pmin:=1001;

num:=-1;

pnum:=-1;

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

**begin**

readln(x);

**if** x>max **then**

**begin**

pmax:=max;

pnum:=num;

num:=i;

max:=x;

**end**

**else if** x>pmax **then**

**begin**

pmax:=x;

pnum:=i;

**end;**

**end;**

**if** pnum<num **then**

writeln(pnum,' ',num)

**else**

writeln(num,' ',pnum);

**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 dfr	2
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

Робота можно разместить на разных гусеничных шасси. Зная массу робота, массу каждого типа шасси и площадь поверхности его гусениц, найдите шасси, которое обеспечит минимальное давление на поверхность. Все шасси отличаются давлением, которое они оказывают.

*Формат ввода*

В первой строке вводится натуральное число  $N$  – количество шасси.  $1 \leq N \leq 15$ . 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, не превосходят 10000 и имеют физический смысл.

*Формат вывода*

Вывести через пробел одно целое число – номер шасси, которое обеспечит минимальное давление на поверхность.

### Пример

Входные данные	Выходные данные
3 10 0.1 0.3 1.5 2 0.6 0.7	2

### Тесты

Входные данные	Выходные данные
3 10 0.1 0.3 1.5 2 0.6 0.7	2
1 0.001 0.001 10000	1
5 10 1 2 2 1 0.5 1 2 3 7 1	4
1 1000 10000 0.1	1

5 10 1 2 1 1.5 4 3 5 2 1 0.7	3
--	---

**Решение**

```
program s8zv1;  
var  
  n,i:integer;  
  ro,h,max,pmax:real;  
begin  
  readln(n);  
  max:=-1;  
  for i:=1 to n do  
    begin  
      readln(ro,h);  
      if ro*h>max then  
        begin  
          max:=ro*h;  
          pmax:=i;  
        end;  
    end;  
  writeln(pmax);  
end.
```