

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

**Вариант 2**

**Задача 1**

Интернет-тролль Ибрагим просыпается в  $h$  часов  $m$  минут и сразу же начинает без остановки строчить сообщения в соцсети, провоцируя пользователей на эмоциональную перепалку с переходом на личности. На написание и отправку каждого сообщения уходит  $n$  секунд.

Юзер Юся просыпается ровно в 8 часов, и тут же начинает потреблять сегодняшние сообщения, тратя по 30 секунд на каждое. Если сообщение тролля уже написано, потребляет именно его.

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

Временем начала перепалки считается время начала написания юзером первого сообщения. Сколько времени будет на часах, когда интернет-тролль Ибрагим достигнет своей цели?

На вход программе в первой строке через пробел подаются 4 целых числа – данные тролля:  $h, m, n$  и  $k$ . Время, когда просыпаются тролль и юзер, указано в одних сутках.  $n, k > 0$ . Исходные данные заданы так, что время начала перепалки 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 перепалки.

**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 0 60 4	8 4 30
4 0 10 1000	16 20 0
8 0 10 1000	16 20 30
8 10 10 1000	16 30 30
17 10 10 800	23 50 30
8 0 60 10	8 10 30
7 0 60 10	8 5 0
7 55 60 10	8 5 30
9 0 35 7	9 5 0

**Решение**

```
hi, mi, ni, k = map(int, input().split())
```

```
hu, mu, nu = 8, 0, 30
```

```
i = (hi * 60 + mi) * 60
```

```
u = (hu * 60 + mu) * 60
```

```

# время окончания написания k сообщений
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 небоскребе  $n$  этажей ( $n < 10^{10}$ ). Известно, что если уронить некий стеклянный шарик с этажа номер  $p$ , и шарик не разобьется, то если уронить шарик с этажа номер  $p-1$ , то он тоже не разобьется. Известно также, что если уронить стеклянный шарик с этажа номер  $p$ , и шарик разобьется, то если уронить шарик с этажа номер  $p+1$ , то он тоже разобьется. Кроме того известно, что при броске с последнего этажа шарик всегда разбивается.

Вы хотите определить минимальный номер этажа, при падении с которого шарик разбивается. Для проведения экспериментов у вас есть два шарика. Вы можете разбить их все, но в результате вы должны абсолютно точно определить этот номер.

Определите, какого числа бросков достаточно, чтобы заведомо решить эту задачу.

### Формат входных данных

Программа получает на вход натуральное число – количество этажей в небоскребе.

### Формат выходных данных

Требуется вывести наименьшее число бросков, при котором можно всегда решить задачу.

### Пример

Входные данные	Выходные данные
2	1

### Тесты

Входные данные	Выходные данные
1	0
2	1

10	4
100	14
93	14
92	13
800000000	40000

### Решение (C++)

```
#include <iostream>
#include <cmath>

int main(void)
{
    int n = 0;
    std::cin >> n;
    std::cout << ceil((-1 + sqrt(1 + 8 * (n - 1))) / 2);
    return 0;
}
```

### 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 двоичного представления  $X$ ;
- вторая цифра числа  $Y$  – это крайняя правая цифра четверичного представления  $X$ ;
- последняя цифра числа  $Y$  – это крайняя правая цифра восьмеричного представления  $X$ .

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

Например,  $X = 187_{10} = 10111011_2 = 2323_4 = 273_8$ . Тогда новое число  $Y = 133_{16} = 307_{10}$ .

Напишите программу, которая будет искать наименьшее 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
185 187	34	185 -> 273 186 -> 34 наименьшее 187 -> 307

## Тесты

Ввод	Вывод
1 7	4
159 159	311
1969 1971	34
1 1000000	0
999990 1000000	0

## Решение

```
program automatic1;
```

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

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

```
begin
```

```
    digit3 := x mod 8;
```

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

```
    digit1 := x mod 2;
```

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

```
end;
```

```
var a, b, n, min, y: longint;
```

```
begin
```

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

```
    min := 100000;
```

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

```
        begin
```

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

```
            writeln(y);
```

```
            if y < min then
```

```
                min := y
```

```
        end;
```

```
    write(min)
```

```
end.
```

## Задача 4

Шарики, каждый из которых объемом  $100 \text{ см}^3$ , бросают в воду и керосин. Зная массу каждого шарика, найти разность массы между самым легким из тех, что тонет в воде и самым тяжелым из тех, что не утонет в керосине. Известно, что среди шариков есть хотя бы один тяжелее воды и хотя бы один легче керосина. Массы всех шариков разные.

*Формат ввода*

В первой строке программы вводится натуральное число  $N$  ( $2 \leq N \leq 30$ ) – количество шариков. Далее в  $N$  строчках вводится по одному натуральному числу  $m_i$  – масса шарика под номером  $i$ . Масса указывается в граммах. Плотность воды –  $1 \text{ г/см}^3$ . Плотность керосина –  $0,8 \text{ г/см}^3$ .

*Формат вывода*

Вывести одно целое число – разницу по массе между самым легким из тех шариков, что тонет в воде и самым тяжелым из тех, что не утонет в керосине.

### Пример

Входные данные	Выходные данные
3 120 99 67	53

### Тесты

Входные данные	Выходные данные
3 120 99 67	53
5 80 79 101 200 56	21
5 80 101 45 150 90	21
3 79 105 56	26
2 80 101	21

### Решение

```
program pzv2;  
var  
  i,n,x,max,min:integer;  
begin  
  readln(n);  
  max:=0;  
  min:=10000;  
  for i:=1 to n do  
  begin  
    readln(x);  
    if (x>max) and (x<=80) then  
      max:=x;  
    if (x<min) and (x>100) then
```

```

min:=x;
end;
writeln(min-max);
end.

```

### Задача 5

Антон, Борис и Владимир передают зашифрованные сообщения в Центр. Сообщения состоят только из строчных латинских букв. В каждом сообщении последовательность букв **qvb** встречается только один раз и является разделителем. В сообщении от Антона в цепочке знаков от начала до разделителя на одну букву больше, чем от разделителя до окончания, от Бориса – на две, а от Владимира – ровно на три. В остальных случаях это сообщение считается шумом. Найдите, кто отправил больше всего сообщений. Такой агент обязательно будет один. Код Антона – 1, код Бориса – 2, код Владимира – 3.

*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 – код агента, который передал больше всего сообщений в Центр.

### Примеры

Входные данные	Выходные данные
3 abcqvba bcfvbe abqvbr	1

### Тесты

Входные данные	Выходные данные
3 abcqvba bcfvbe abqvbr	1
7 avqqqvbd aqvb aqvb aaqv aaqv aqvb avqqqvbd	1
7 aaaqv aaaqv ayaqvbuia ayutqv qvba eaqvbraa yuyyqvbyay	3
5	3

qvb qvb qvba qvba aaaqvb	
1 aqvb	1

### Решение

```
program szv3;
```

```
function delta(s,s1:string):integer;
```

```
var
```

```
l:integer;
```

```
begin
```

```
l:=length(copy(s,1,pos(s1,s)-1));
```

```
delete(s,1,pos(s1,s)+length(s1)-1);
```

```
delta:=1 - length(s);
```

```
end;
```

```
var
```

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

```
s:string;
```

```
begin
```

```
readln(n);
```

```
a:=0;
```

```
b:=0;
```

```
c:=0;
```

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

```
  begin
```

```
    readln(s);
```

```
    if delta(s,'qvb')=1 then
```

```
      a:=a+1;
```

```
    if delta(s,'qvb')=2 then
```

```
      b:=b+1;
```

```
    if delta(s,'qvb')=3 then
```

```
      c:=c+1;
```

```
    end;
```

```
  if (a>b) and (a>c) then
```

```
    writeln(1);
```

```
  if (b>a) and (b>c) then
```

```
    writeln(2);
```

```
  if (c>a) and (c>b) then
```

```
    writeln(3);
```

```
end.
```

### Задача 6 Ситуационная задача

В сосуд по очереди заливают разные жидкости. Зная давление жидкости и высоту ее уровня относительно дна сосуда, найдите наименее плотную жидкость. Все жидкости различаются по плотности.

#### Формат ввода

В первой строке программы вводится натуральное число  $N$  – количество видов жидкости,  $1 \leq N \leq 20$ . Далее в  $N$  строчках вводится через пробел по два положительных вещественных числа  $P_i$  и  $h_i$  –

давление на дно сосуда жидкости номер  $i$  и высота ее уровня, соответственно. Все величины приведены в системе СИ, не превосходят 10000 и имеют физический смысл.

*Формат вывода*

Вывести одно целое число – номер жидкости с наименьшей плотностью.

### Пример

Входные данные	Выходные данные
5 1500 1 2000 3 1555 1.3 898.8 0.99 9000 5.2	2

### Тесты

Входные данные	Выходные данные
5 1500 1 2000 3 1555 1.3 898.8 0.99 9000 5.2	2
1 1 1	1
3 1000 1 2000 3 5000 6000	3
5 1 2 2 3 3 4 1 0.1 5 11	5
1 1000 1	1



## Решение

```
program s8zv2;  
var  
  n,i,nmin:integer;  
  p,h,min:real;  
begin  
  readln(n);  
  nmin:=1;  
  readln(p,h);  
  min:=p/h;  
for i:=2 to n do  
  begin  
    readln(p,h);  
    if p/h<min then  
    begin  
      min:=p/h;  
      nmin:=i;  
    end;  
  end;  
  writeln(nmin);  
end.
```