

**Отборочный (заочный) онлайн-этап академического соревнования Олимпиады школьников  
«Шаг в будущее» по программированию «Профессор Лебедев»  
(общеобразовательный предмет информатика), осень 2020 год**

**8 класс**

**Вариант 2**

**Задача 1**

Пётр Васильевич работает учителем информатики и каждый рабочий день ездит в школу. До школы он может добраться на трамвае, который отправляется ходит с одинаковым интервалом в 16 минут. Первый трамвай идёт в 5 часов 30 минут. Сегодня Пётр Васильевич подошёл к остановке в  $h$  часов  $m$  минут и хочет уехать как можно скорее. Если бы в ту же минуту подошёл его трамвай, он бы уехал сразу. Напишите программу, которая определит, во сколько сегодня уедет Пётр Васильевич. На вход программе в одной строке подаются два целых числа, отделенных друг от друга одним пробелом:  $h$  ( $5 < h \leq 18$ ) и  $m$  ( $0 \leq m < 60$ ).

В одной строке выходных данных через двоеточие программа должна вывести искомое время: час и минуту отправления. Минуты выводить в виде двух цифр.

**Пример**

Входные данные	Выходные данные
7 2	7:06

**Пояснение** **к** **примеру**  
Трамвай отправляется в 5:30, 5:46, 6:02, 6:18, 6:34, 6:50, 7:06, 7:22, ....  
Придя на остановку в 7:02, Пётр Васильевич уедет на трамвае в 7:06.

**Тесты**

Входные данные	Выходные данные
7 2	7:06
10 0	10:02
10 1	10:02
10 2	10:02
10 3	10:18
6 0	6:02
18 59	19:06
15 53	15:54
15 54	15:54
15 55	16:10
16 0	16:10

**Решение**

**const**

$t0 = 5 * 60 + 30;$

$d = 16;$

**var**

$h, m, dt, t: \text{longint};$

**Begin**

$\text{read}(h, m);$

$dt := (h * 60 + m) - t0;$

$t := (dt - 1) \text{ div } d * d + d + t0;$

$m := t \text{ mod } 60;$

$h := t \text{ div } 60;$

$\text{writeln}(h, ':', m \text{ div } 10, m \text{ mod } 10)$

**End.**

### Задача 2

Некоторое устройство получает на вход положительное целое число и строит по нему новое следующим образом: записываются все цифры десятичного представления исходного числа в обратном порядке, а затем удаляются лидирующие нули.

Определите количество различных чисел  $n$  ( $1 \leq n \leq x$ ), которые можно получить разделив исходное число на число, полученное после двойного применения описанного алгоритма для числа  $x$ .

Входные данные

Первая строка содержит одно целое число  $n$  ( $1 \leq x < 18446744073709551615$ ). Это целое число задается без лидирующих нулей.

Выходные данные

Выведите одно целое число — количество различных значений, которые может принимать  $g(x)$ , если  $x$  может быть любым целым числом из  $[1, n]$ .

### Пример

Входные данные	Выходные данные
4	1

### Тесты

Входные данные	Ожидаемый результат
4	1
37	2
123456789	9
100	3
998244353	9
1000000007	10
12345678901337426	17

### Решение

```
python 3
n = int(input())
for i in range(n):
    print(len(input()))
```

### Задача 3

Элемент последовательности называется локальным максимумом, если он строго больше предыдущего и последующего элемента последовательности. Первый и последний элемент последовательности являются локальными максимумами, если они больше своего единственного соседнего элемента.

Дано натуральное число  $N$  ( $1 < N \leq 10^9$ ). Напишите программу, которая подсчитывает локальных максимумов в последовательности цифр двоичного представления данного числа.

На вход программы подаётся число  $N$ .

Программа должна вывести одно целое число - вычисленное значение.

### Пример

Ввод	Вывод	Примечание
53	2	$53_{10} = 110101_2$ 1-ый максимум <u>110</u> <b>1</b> 01 2-ой максимум 1101 <b>0</b> <u>1</u>

24	0	$24_{10} = 11000_2$
----	---	---------------------

### Тесты

Ввод	Вывод
2	1
5	2
10	2
13	1
107	1
32768	1
43690	8
54613	7
65535	0
1000000000	3

### Решение

```

program tur1_2020_2021;
var n: longint;
count, digit, preDigit, nextDigit: integer;
begin
  read(n);
  digit := n mod 2;
  n := n div 2;
  if digit > n mod 2 then
    count := 1
  else
    count := 0;
  while n > 0 do
    begin
      preDigit := digit;
      digit := n mod 2;
      n := n div 2;
      nextDigit := n mod 2;
      if (digit > preDigit) and (digit > nextDigit) then
        count := count + 1;
    end;
  write(count)
end.

```

### Задача 4

Для сборки конструкции нужны две балки. Зная массу каждой балки, найти пару балок с наибольшей суммарной массой. Две любые балки отличаются по массе.

*Формат ввода*

В первой строке программы вводится натуральное число  $N$  – количество балок ( $1 < N < 100$ ). Далее в  $N$  строчках вводится по одному положительному вещественному числу  $m_i$  – масса балки под номером  $i$ .

*Формат вывода*

Вывести через пробел два целых числа – номера балок, суммарная масса которых максимальна. Сначала вывести наименьший из номеров.

### Пример

Входные данные	Выходные данные
5	1 3

5.2 3.9 4.1 1.1 2.0	
---------------------------------	--

### Тесты

Входные данные	Выходные данные
5 5 3 4 1 2	1 3
2 1 2	1 2
2 2 1	1 2
10 6 7 15 23 45 43 27 19 31 33	5 6
7 98 1 2 4 5 7 99	1 7

### Решение

```
program pzv1;
```

```
var
```

```
i,n,num,pnum:integer;
```

```
x,max,pmax:real;
```

```
begin
```

```
  readln(n);
```

```
  max:=-1;
```

```
  pmax:=-1;
```

```
  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

Николай, Сергей и Максим передают зашифрованные сообщения в Центр. Сообщения состоят только из строчных латинских букв. Сообщение от Николая обязательно содержит ровно одну букву **a**, от Сергея – ровно две буквы **a**, а от Максима – ровно три буквы **a**. Если букв **a** больше трех или ноль, это сообщение считается шумом. Найдите, кто отправил больше всего сообщений. Такой агент обязательно будет один. Код Николая – 1, код Сергея – 2, код Максима – 3.

*Формат ввода*

В первой строке вводится сначала натуральное число  $n$  – количество сообщений ( $n \leq 1000$ ), 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
6 aaa asdf asdrt baa ama acs	1

### Тесты

Входные данные	Выходные данные
3 avc aab bsa	1
6 aaa	1

asdf asdr baa ama acs	
7 aaa aaa ayauia ayu auy earaa yyyyay	3
5 b b b b aa	2
1 aaa	3

### Решение

**program** szv3;

**function** calc(s,s1:string):integer;

**var**

i,k:integer;

**begin**

k:=0;

**for** i:=1 **to** length(s) **do**

**begin**

**if** copy(s,i,1)=s1 **then**

k:=k+1;

**end;**

calc:=k;

**end;**

**var**

n,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** calc(s,'a')=1 **then**

a:=a+1;

```

if calc(s,'a')=2 then
  b:=b+1;
if calc(s,'a')=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 браузерной игре, посвященной Боям космических кораблей, игрок каждый ход может использовать одну из пяти тактик. Во время боя случайные события могут помешать применить какую-то одну из тактик.

Сколько существует последовательностей тактических ходов длиной  $n$ , если известна цепочка случайных событий, произошедших за эти ходы?

На вход программе подается строка длиной  $n$  ( $N \leq 20$ ) символов, состоящая из цифр 0,1,2,3,4,5.

Символ 0 означает, что случайных событий на этом ходу нет.

Символы с 1 по 5 означают, что нельзя задействовать тактику с соответствующим номером.

Вывести целое число – количество последовательностей тактических ходов.

### Пример

Ввод	Вывод
11	16

### Тесты

Ввод	Вывод
1	4
00000	3125
11	16
12345	1024
00001230	200000

### Решение

```

program z10083;

```

```

function sum(p:integer; s1,sk:string):integer;

```

```

begin

```

```

  if s1=sk then

```

```

    sum := 0

```

```

  else

```

```

    sum:=p;

```

```

end;

```

```

var

```

```

  s,s1:string;

```

```

  i,a,b,c,d,e,psum:longint;

```

```

begin

```

```

  readln(s);

```

```
psum:=1;
s1:=copy(s,1,1);
a:=sum(psum,s1,'1');
b:=sum(psum,s1,'2');
c:=sum(psum,s1,'3');
d:=sum(psum,s1,'4');
e:=sum(psum,s1,'5');
for i:=2 to length(s) do
  begin
    s1:=copy(s,i,1);
    psum:=a+b+c+d+e;
    a:=sum(psum,s1,'1');
    b:=sum(psum,s1,'2');
    c:=sum(psum,s1,'3');
    d:=sum(psum,s1,'4');
    e:=sum(psum,s1,'5');
  end;
  writeln(a+b+c+d+e);
end.
```