

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации
«Техника и технологии» (общеобразовательный предмет информатика), весна 2019 г.
8 класс**

Вариант 3

Задача 1.

Петя скоро пойдёт в школу, и он влюблён в числа. Недавно ему рассказали, как считать после девяти. Особенно Пете нравились числа, которые заканчивались на две одинаковые цифры. Он принялся их считать.

Напишите программу, которая проверит Петины подсчёты.

На вход программы подаётся два числа A и B , записанные через пробел ($1 \leq A \leq B \leq 10^9$).

Программа должна вывести количество чисел от A до B (включая сами числа A и B), которые так понравились Пете.

Пример

Входные данные	Выходные данные
190 210	2

Комментарий к примеру:

от 190 до 210 включительно есть 2 искомых числа: 199 и 200.

Решение

```
function nice(n: longint): boolean;
var s: longint;
begin
  s := 0;
  while n > 0 do begin
    inc(s, n mod 10);
    n := n div 10;
  end;
  nice := s mod 2 <> 0;
end;

function count(a, b: longint): longint;
var res, i: longint;
begin
  res := 0;
  for i := a to b do
    if nice(i) then
      inc(res);
  count := res;
```

end;

var a, b, k, a1, b1: longint;

Begin

read(a, b);

k := 0;

if b >= a then begin

a1 := min(b, (a div 10 + 1) * 10 - 1); // конец текущего десятка A - aaa9

inc(k, count(a, a1));

a := a1 + 1;

if b >= a then begin

b1 := max(a, b div 10 * 10); // последний десяток bbb0 - xxxb

inc(k, count(b1, b));

b := b1 - 1;

if b >= a then

inc(k, 5 * (b + 1 - a) div 10);

end;

end;

write(k)

End.

Тесты

N	Оценка	Входные данные	Выходные данные
1	2	222 999	78
2	2	222 998	77
3	2	223 999	77
4	1	1 10	0
5	3	10 999999000	99999900

Задача 2.

Будем считать подходящим десятичное число, шестнадцатеричная запись которого содержит не менее X значащих цифр. Сколько таких чисел существует в диапазоне от M до N включительно?

Формат ввода

В строке вводятся три натуральных числа M, N, X ,

$M \leq N \leq 4000000$,

$X \leq 6$

Формат вывода

Вывести одно целое число – количество целых чисел, удовлетворяющих условию. Если таких чисел нет, вывести 0.

Пример

Входные данные	Выходные данные
5 30 2	15

Решение

```

program my;
const
  O = 16;

function intpower(x:longint):longint;
var i,c:integer;
begin
  c:=1;
  for i:=1 to x do
    c:=c*O;
  intpower:=c;
end;

var i,s,c,m,n,x:longint;
begin
  read(m,n,x);
  s:=0;
  c:=intpower(x-1);
  for i:=m to n do
    if i>=c then
      s:=s+1;
  writeln(s);
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	3	30 60 2	21
2	3	256 256 1	1
3	3	20 40 1	21
4	3	10 50000 3	49745
5	3	1 4000000 8	2951425

Задача 3.

Есть исполнитель **Сдвиг**. Он преобразует целое неотрицательное число, которое хранится в одном байте, по следующему правилу:

1) Если количество единиц в двоичной записи этого числа нечётное, то смещаем все двоичные цифры в байте на 2 позиции влево и записываем на образовавшиеся место два бита со значением **00**.

Например,

19_{10}

00010011_2

Преобразуется в

$010011_ _$ (сдвиг влево)

01001100_2 (результат)

76_{10}

2) Если количество единиц в двоичной записи этого числа чётное, то смещаем все двоичные цифры в байте на **1** позицию вправо и записываем на образовавшиеся место бит со значением **1**, если при этом количество единиц в двоичной записи стало нечетным, и **0**, если количество единиц осталось чётным.

Например,

20_{10}

00010100_2

Преобразуется в

$_0001010$ (сдвиг вправо)

00001010_2 (результат)

10_{10}

Напишите программу, которая осуществляет данное преобразование над введенным целым неотрицательным числом.

На вход программы подаётся число N .

Программа должна вывести одно целое число - вычисленное значение.

Пример

Ввод	Вывод
20	10
19	76

Решение

```
function f(n:integer):integer;
```

```
var s,x:integer;
```

```
begin
```

```
x:=n;
```

```
s:=0;
```

```
while x>0 do
```

```
begin
```

```
s:=s+ x mod 2;
```

```
x:= x div 2;
```

```
end;
```

```
if s mod 2 =0 then
```

```
f:= n div 2+ (n mod 2)*128
```

```
else
```

```
f:=(n*4) mod 256
```

```
end;
```

```
var n:integer;
```

```
begin
```

```
readln(n) ;
```

```
writeln(f(n))
```

```
end.
```

N	Оценка	Входные данные	Выходные данные
1	1	0	0

2	1	1	4
3	1	2	8
4	1	22	88
5	1	39	147
6	2	79	60
7	2	102	51
8	2	128	0
9	2	192	96
10	2	255	255

Задача 4.

Суперагентам Буке и Бяке необходимо получить кодовый ключ к замку. Для того, чтобы составить ключ необходимо записать в одну строку без пробелов все слова, составленные из букв некоторого алфавита, буквы в них идут в алфавитном порядке, но повторяться могут от 1 до N раз. Слова можно записать в любом порядке.

В качестве кода требуется набрать два числа: длину строки и количество слов в строке.

Входные данные:

На вход программе подаётся строка, содержащая алфавит, из которого надо составлять слова, в виде последовательности символов, записанных без пробела и число N ($N \leq 10$), записанное через пробел после алфавита. В качестве буквы алфавита может быть использован любой символ. Количество букв в алфавите не превышает 10.

Выходные данные:

Два числа, записанные через один пробел. Первое число - количество слов, второе - длина строки.

Пример:

Входные данные	Выходные данные
AB 2	4 12

Пояснение:

Получится составить следующие 6 слов максимальной длины:

AB ABV AAB AABV

Решение

```
type ansistring = string;
```

```
function count_letter(const symbol, wrd: ansistring): longint;
```

```
var i, amount: longint;
```

```
begin
```

```
  amount := 0;
```

```
  for i := 1 to length(wrd) do
```

```
    if copy(wrd, i, 1) = symbol then
```

```
      amount := amount + 1;
```

```

    count_letter := amount;
end;
function doubl(s: string; n: longint): string;
var res: string;
    i: longint;
begin
    res:= "";
    for i := 1 to n do
        res := res + s;
        doubl := res;
    end;
end;

```

```

procedure count_and_build_words(N: longint; const alphabet: ansistring; const wrd: ansistring; var
amount, all_length: longint);
var i: longint;
    is_end: boolean;
begin
    if alphabet = " then begin
        amount := amount + 1;
        all_length := all_length + length(wrd);
    end
    else
        for i := 1 to n do
            count_and_build_words(n, copy(alphabet, 2, length(alphabet) - 1), wrd + doubl(copy(alphabet, 1,
1),i), amount, all_length);
        end;
    end;

```

```

function fact(f: int64): int64;
var i, p: int64;
begin
    p := 1;
    for i := f downto 1 do
        p := p * i;
    fact := p;
end;

```

```

procedure count_and_build_words_fast(n: longint; const alphabet: ansistring; const wrd: ansistring; var
amount2, all_length2: longint);
var wrd_len, i: longint;
    is_end: boolean;
    amount, all_length: int64;
begin
    amount2 := round(power(n, length(alphabet)));
    all_length2 := amount2 * (length(alphabet) + trunc((length(alphabet) / 2) * (n-1)));
end;

```

```

var
    alphabet: ansistring;
    amount, all_length, n: longint;
begin
    read(alphabet);
    readln(n);

```

```

amount := 0;
all_length := 0;
count_and_build_words(n,alphabet, "", amount, all_length);
amount := 0;
all_length := 0;
count_and_build_words_fast(n, alphabet, "", amount, all_length);
writeln(amount, ' ', all_length);
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	3	Abc 5	125 1125
2	5	Abcde 1	1 5
3	5	Abcd 8	4096 73728
4	7	Abcdefg 10	10000000 385000000
5	10	Abcdefghij 6	60466176 2116316160