

**Заключительный (очный) этап академического соревнования  
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.  
8 класс**

**Вариант 1**

**Задача 1.**

Вася знал номер квартиры  $n$ , где живёт его одноклассник, но совершенно не помнил этаж. На табличке над входом в подъезд было написано, что первая квартира имеет номер  $n_1$ . Войдя в подъезд, мальчик увидел, что на этаже расположено  $k$  квартир. Напишите программу, которая поможет Васе определить этаж, на котором живёт его одноклассник, исходя из имеющейся информации.

На вход программы подаются три числа  $n$ ,  $n_1$  и  $k$ , записанные через один пробел, каждое из которых не превышает тысячи.

Программа должна вывести одно число – номер этажа.

**Пример**

Входные данные	Выходные данные
140 131 4	3

**Решение**

```
var n, n1, k: longint;
```

```
Begin
```

```
  readln(n, n1, k);
```

```
  writeln((n - n1) div k + 1)
```

```
End.
```

N	Оценка	Входные данные	Выходные данные
1	1	123 12 8	14
2	1	124 12 8	15
3	2	1000 1000 1000	1
4	1	5 1 3	2

**Задача 2.**

Петя скоро пойдёт в школу, и он влюблён в числа. Недавно ему рассказали, как считать после девяноста девяти. Особенно Пете нравились числа, которые начинались и заканчивались на одну и ту же цифру. 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*

$k$

*примеру:*

от 190 до 210 включительно Пете особенно понравятся два числа: 191 и 202.

### Решение

```
function senior(n: longint): longint;
```

```
begin
```

```
  while n > 9 do
```

```
    n := n div 10;
```

```
  senior := n;
```

```
end;
```

```
var a, b, i, k: longint;
```

```
Begin
```

```
  read(a, b);
```

```
  k := 0;
```

```
  if a < 10 then begin
```

```
    inc(k, 10 - a);
```

```
    a := 10;
```

```
  end;
```

```
  if a mod 10 > senior(a) then
```

```
    dec(k);
```

```
  a := a div 10 * 10;
```

```
  if b mod 10 < senior(b) then
```

```
    dec(k);
```

```
  b := b div 10 * 10 + 10;
```

```
  inc(k, (b - a) div 10);
```

```
  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 3.

Будем считать подходящим десятичное число, восьмеричная 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  $M, N, X$ ,

$M \leq N \leq 4000000$ ,

$X \leq 9$

*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	23

### Решение

```
program my;
const
  O = 8;

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 3	0
2	3	64 64 3	1

3	3	20 40 1	21
4	3	10 50000 6	17233
5	3	1 4000000 8	1902849

#### Задача 4.

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 **01**.

Например,

$19_{10}$

$00010011_2$

Преобразуется в

$010011\_ \_$  (сдвиг влево)

$01001101_2$  (результат)

$77_{10}$

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**.

Например,

$18_{10}$

$00010010_2$

Преобразуется в

$\_0001001$  (сдвиг вправо)

$10001001_2$  (результат)

$137_{10}$

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

На вход программы подаётся число  $N$ .

Программа должна вывести одно целое число - вычисленное значение.

#### Примеры

Ввод	Вывод
19	77
18	137

## 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:=128+n div 2
else
f:=1+ (n * 4) mod 256
end;

var n:integer;
begin
readln(n);
writeln(f(n))
end.
```

N	Оценка	Входные данные	Выходные данные
1	1	0	128
2	1	1	1
3	1	2	9
4	1	22	89
5	1	39	147
6	2	79	61
7	2	102	179
8	2	128	1
9	2	192	224
10	2	255	255

### Задача 5.

Прототип робота-уборщика исполняет четыре команды: «вперед», «стоп», «поворот налево» и «поворот направо». Поскольку это робот-прототип, он не может поворачивать на ходу; перед поворотом надо отдать команду «стоп». Сколько разных выполнимых последовательностей можно составить из заранее известного количества команд, если в начальный момент времени робот стоит?

*Формат ввода*

В строке вводится сначала натуральное число  $n$  – количество команд ( $n \leq 30$ ).

*Формат вывода*

Вывести одно целое число – количество способов, которыми можно составить последовательности команд длиной  $n$  для робота.

### Пример

Входные данные	Выходные данные
2	14

### Решение

```

program my;
var n,i,a,b,c,d,pa,pb,pc,pd:longint;
begin
  pa:=0;
  pb:=1;
  pc:=0;
  pd:=0;
  readln(n);
  for i:=1 to n do
  begin
    a:=pa+pc+pb+pd;
    b:=pa+pd+pb+pc;
    c:=pb+pc+pd;
    d:=pb+pc+pd;
    pa:=a;
    pb:=b;
    pc:=c;
    pd:=d;
  end;
  writeln(a+b+c+d);
end.

```

### Тесты

N	Оценка	Входные данные	Выходные данные
1	5	30	1818198016
2	5	3	48
3	5	1	4
4	5	10	259808
5	5	8	22288

### Задача 6.

Суперагентам Маше и Вите необходимо получить кодовый ключ к замку. Для того, чтобы составить ключ необходимо записать в одну строку без пробелов все слова максимально возможной длины, составленные из букв некоторого алфавита, в которых каждая буква повторяется не более  $N$  раз ( $N$  – порядковый номер этой буквы в алфавите). 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 качестве буквы алфавита может быть использован любой символ. Количество букв в алфавите не превышает 5.

### **Выходные данные:**

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

### **Пример:**

Входные данные	Выходные данные
<b>AB</b>	<b>3 9</b>

*Пояснение:*

В приведённом алфавите буква **A** стоит на 1 месте, а буква **B** - на втором. Соответственно буква **A** может присутствовать в слове не более 1 раза, а буква **B** – не более 2 раз. Тогда получится составить следующие 3 слова максимальной длины: **ABB BAV BBA**

### **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;

procedure count_and_build_words(const alphabet: ansistring; const wrd: ansistring; var amount,
all_length: longint);
var i: longint;
    is_end: boolean;
begin
    is_end := true;
    for i := 1 to length(alphabet) do
        if count_letter(copy(alphabet, i, 1), wrd) < i then begin
            is_end := false;
            count_and_build_words(alphabet, wrd + copy(alphabet, i, 1), amount, all_length);
        end;
    if is_end then begin
        //writeln(wrd);
        amount := amount + 1;
        all_length := all_length + length(wrd);
    end
end;
```

```

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

procedure count_and_build_words_fast(const alphabet: ansistring; const wrd: ansistring; var amount,
all_length: longint);
var wrd_len, i: longint;
  is_end: boolean;
begin
  wrd_len := 0;
  for i := 1 to length(alphabet) do
    wrd_len := wrd_len + i;

  amount := 1;
  all_length := wrd_len;
  for i := 1 to length(alphabet) do begin
    amount := amount * fact(all_length) div fact(all_length - i) div fact(i);
    all_length := all_length - i;
  end;

  all_length := amount * wrd_len;
end;

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

  { amount := 0;
  all_length := 0;
  count_and_build_words(alphabet, "", amount, all_length);
  writeln(amount, ' ', all_length); }

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

```

### Тесты

N	Оценка	Входные данные	Выходные данные
1	3	BA	3 9
2	5	ABC	60 360
3	5	CBA	60 360
4	7	ABDC	12600 126000
5	10	ABCDE	27720 415800

**Заключительный (очный) этап академического соревнования  
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.  
8 класс**

**Вариант 2**

**Задача 1.**

Андрей обожает рельсовый транспорт. В кармане он всегда носит свой любимый трамвай *Татра Т3*, игрушечный, конечно. В очередной раз он упросил маму не ехать на машине, а ехать на метро. В метро Андрей попросил маму подождать, пока он не научится предсказывать появление поезда. На часах было ровно  $h$  часов и  $m$  минут, когда показался первый поезд. Помогите Андрею рассчитать, в какой момент появится  $n$ -ый поезд, если интервал движения поездов составляет  $dm$  минут,  $ds$  секунд.

На вход программы подаются пять чисел:  $h$ ,  $m$ ,  $n$  и  $dm$  с  $ds$ , записанные через один пробел.

Программа должна вывести через пробел два целых числа без ведущих нулей: сколько часов и минут должно быть на часах метрополитена, исходя из имеющихся данных.

**Пример**

Входные данные	Выходные данные
18 00 4 2 30	18 7

*Комментарий к примеру:*

1-ый поезд – 18:00:00, 2-ой – 18:02:30, 3-ий – 18:05:00, 4-ый – **18:07:30**.

**Решение**

```
var h, m, n, dm, ds, t: longint;
Begin
  read(h, m, n, dm, ds);
  t := (h * 60 + m) * 60 + (n - 1) * (dm * 60 + ds);
  t := t div 60;
  write(t div 60 mod 24, ' ', t mod 60);
End.
```

**Тесты**

N	Оценка	Входные данные	Выходные данные
1	1	23 50 1 3 20	23 50
2	1	23 50 3 3 20	23 56
3	2	23 50 4 3 20	0 0
4	1	0 0 5000 1 1	12 42

## Задача 2.

Петя скоро пойдёт в школу, и он влюблён в числа. Недавно ему рассказали, как считать после девяти. Особенно Пете нравились числа, у которых нечётная сумма цифр. 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
20 30	6

*Комментарий к примеру:*

от 20 до 30 есть 6 искомым чисел: 21, 23, 25, 27, 29 и 30.

### 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	1	99 99	0
2	2	99 300	101
3	2	999 3000	1002
4	2	999 1001	2
5	3	15 999999985	70503255

### Задача 3.

Будем считать подходящим десятичное число, двоичная запись которого заканчивается не менее, чем  $X$  нулями. Сколько таких чисел существует в диапазоне от  $M$  до  $N$  включительно?

*Формат ввода*

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

$M \leq N \leq 4000000$ ,

$X \leq 20$

*Формат вывода*

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

### Пример

Входные данные	Выходные данные
2 30 3	3

### Решение

```

program my;

```

```

const

```

```

O = 2;

```

```

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);
for i:=m to n do
  if i mod c = 0 then
    s:=s+1;
  writeln(s);
end.

```

### Тесты

N	Оценка	Входные данные	Выходные данные
1	3	<b>60 60 2</b>	1
2	3	<b>30 63 6</b>	0
3	3	<b>2 28 1</b>	14
4	3	<b>100 500 6</b>	6
5	3	<b>1 4000000 8</b>	15625

### Задача 4.

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 вправо и записываем на образовавшиеся место два бита со значением **10**.

Например,

**19**<sub>10</sub>

**00010011**<sub>2</sub>

Преобразуется в

**\_\_ 000100** (сдвиг вправо)

**10000100**<sub>2</sub> (результат)

**132**<sub>10</sub>

2) Если количество единиц в двоичной записи этого числа чётное, то смещаем все двоичные цифры в байте на **2** позиции влево и записываем на образовавшиеся место два бита со значением **10**.

Например,

**12**<sub>10</sub>

$00001100_2$

Преобразуется в

$001100\_ \_$  (сдвиг влево)

$00110010_2$  (результат)

$50_{10}$

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

На вход программы подаётся число  $N$ .

Программа должна вывести одно целое число - вычисленное значение.

### Пример

Ввод	Вывод
19	132
12	50

### 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:=2+(n*4) mod 256
else
f:=128+(n div 4)
end;

var n:integer;
begin
readln(n) ;
writeln(f(n))
end.
```

N	Оценка	Входные данные	Выходные данные
1	1	0	2
2	1	1	128
3	1	2	128
4	1	22	133
5	1	39	158
6	2	79	147

7	2	102	154
8	2	128	160
9	2	192	2
10	2	255	254

### Задача 5.

Захват манипулятора способен выполнить три команды: «взять», «держать» и «выпустить». Поскольку захват разрабатывали студенты-троичники, он сделан кое-как и при попытке два раза подряд выполнить команду «взять» или «выпустить» ломается. В начальный момент времени захват выполнил команду «выпустить». Сколько можно составить разных последовательностей команд, длина которых не превышает заданное число  $n$ , так, чтобы захват не сломался?

*Формат ввода*

В строке вводится сначала натуральное число  $n$  – количество команд ( $n \leq 15$ ).

*Формат вывода*

Вывести одно целое число – количество способов, которыми можно составить программы длиной не более  $n$  для захвата.

### Пример

Входные данные	Выходные данные
2	7

### Решение

```

program my;
var n,i,a,b,c,d,pa,pb,pc,sum:longint;
begin
pa:=0;
pb:=0;
pc:=1;
sum:=0;
readln(n);
for i:=1 to n do
begin
a:=pb+pc;
b:=pa+pb+pc;
c:=pb+pa;
pa:=a;
pb:=b;
pc:=c;
sum:=sum+a+b+c;
end;
writeln(sum);
end.

```

## Тесты

N	Оценка	Входные данные	Выходные данные
1	5	<b>15</b>	803759
2	5	<b>3</b>	19
3	5	<b>1</b>	2
4	5	<b>10</b>	9799
5	5	<b>8</b>	1680

### Задача 6.

Суперагентам Матильде и Виттеру необходимо получить кодовый ключ к замку. Для того, чтобы составить ключ необходимо записать в одну строку без пробелов все слова максимально возможной длины, составленные из букв некоторого алфавита, в которых каждая буква повторяется не более  $N$  раз ( $N \leq 5$ ). 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  $N$ , записанное через пробел после алфавита. В качестве буквы алфавита может быть использован любой символ. Количество букв в алфавите не превышает 5.

#### **Выходные данные:**

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

#### **Пример:**

Входные данные	Выходные данные
<b>AB 2</b>	<b>6 24</b>

*Пояснение:*

Получится составить следующие 6 слов максимальной длины: **AABB ABAB ABBA BAAB**

**BAVA BBAА**

#### **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;
```

```

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
    is_end := true;
    for i := 1 to length(alphabet) do
        if count_letter(copy(alphabet, i, 1), wrd) < N then begin
            is_end := false;
            count_and_build_words(n, alphabet, wrd + copy(alphabet, i, 1), amount, all_length);
        end;
    if is_end then begin
        amount := amount + 1;
        all_length := all_length + length(wrd);
    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
    wrd_len := length(alphabet) * n;
    amount := 1;
    all_length := wrd_len;
    for i := 1 to length(alphabet) do begin
        amount := amount * fact(all_length) div fact(all_length - n) div fact(n);
        all_length := all_length - n;
    end;
    all_length := amount * wrd_len;
    amount2 := amount;
    all_length2 := all_length;
end;
var
    alphabet: ansistring;
    amount, all_length, n: longint;
begin
    read(alphabet);
    readln(n);
    amount := 0;
    all_length := 0;
    count_and_build_words_fast(n, alphabet, "", amount, all_length);
    writeln(amount, ' ', all_length);
end.

```

## Тесты

№	Оценка	Входные данные	Выходные данные
1	3	AB 5	252 2520
2	5	ABC 2	90 540
3	5	ABCD 2	2520 20160
4	7	ABC 5	756756 11351340
5	10	ABCD 4	63063000 1009008000

**Заключительный (очный) этап академического соревнования  
Олимпиады школьников «Шаг в будущее» по программированию, весна 2019 г.  
8 класс**

**Вариант 3**

**Задача 1.**

Андрей обожает рельсовый транспорт. В кармане он всегда носит свой любимый трамвай *Татра Т3*, игрушечный, конечно. В очередной раз он упросил маму не ехать на машине, а ехать на метро. В метро Андрей попросил маму подождать, пока он не научится предсказывать появление поезда. На часах было ровно  $h_0$  часов и  $m_0$  минут, когда показался первый поезд. Андрей наблюдал за прибытием поездов и, когда на часах было  $h_1$  часов и  $m_1$  минут, показался очередной поезд. Какой по счёту это было поезд, если интервал движения поездов составляет  $dm$  минут,  $ds$  секунд.

На вход программы подаются шесть чисел:  $h_0, m_0, h_1, m_1, dm$  ( $dm > 0$ ) и  $ds$ , записанные через один пробел. Момент времени  $h_1, m_1$  не раньше момента времени  $h_0, m_0$ , но в тех же сутках.

Программа должна вывести через одно число – ответ задачи.

**Пример**

Входные данные	Выходные данные
18 00 18 7 2 30	4

*Комментарий к примеру:*

1-ый поезд – 18:00:00, 2-ой – 18:02:30, 3-ий – 18:05:00, 4-ый – **18:07:30**.

**Решение**

```
var h0, m0, h1, m1, dm, ds, t, n: longint;
Begin
  read(h0, m0, h1, m1, dm, ds);
  t := (h1 * 60 + m1) * 60 + 59 - (h0 * 60 + m0) * 60;
  n := t div (dm * 60 + ds);
  write(n + 1);
End.
```

**Тесты**

N	Оценка	Входные данные	Выходные данные
1	1	10 50 11 00 3 20	4
2	1	0 0 1 25 1 7	78
3	2	0 0 1 1 1 0	62
4	1	10 00 11 02 01 01	62

## Задача 2.

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 искомым 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;
  end;
end;
```

```

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 3.

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	<i>30 60 2</i>	21
2	3	<i>256 256 1</i>	1
3	3	<i>20 40 1</i>	21
4	3	<i>10 50000 3</i>	49745
5	3	<i>1 4000000 8</i>	2951425

### Задача 4.

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<sub>10</sub>*

*00010011<sub>2</sub>*

Преобразуется в

*010011\_ \_* (сдвиг влево)

*01001100<sub>2</sub>* (результат)

*76<sub>10</sub>*

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 5.

Прототип манипулятора исполняет четыре команды: «вперед», «назад», «опустить захват» и «поднять захват». Поскольку это робот-прототип, он не может двигаться с опущенным захватом; перед движением требуется поднять захват. Команда «опустить захват» при опущенном захвате и «поднять захват» при поднятом захвате не может быть выполнена. Сколько разных выполнимых последовательностей можно составить из заранее известного количества команд, если в начальный момент времени робот стоит с поднятым захватом?

*Формат ввода*

В строке вводится сначала натуральное число  $n$  – количество команд ( $n \leq 15$ ).

*Формат вывода*

Вывести одно целое число – количество способов, которыми можно составить последовательности команд длиной  $n$  для робота.

### Пример

Входные данные	Выходные данные
2	7

### Решение

```

program my;
var n,i,a,b,c,d,pa,pb,pc,pd:longint;
begin
pa:=0;
pb:=0;
pc:=0;
pd:=1;
readln(n);
for i:=1 to n do
begin
a:=pa+pb+pd;
b:=pa+pd+pb;
c:=pa+pd+pb;
d:=pc;
pa:=a;

```

```

pb:=b;
pc:=c;
pd:=d;
end;
writeln(a+b+c+d);
end.

```

### Тесты

N	Оценка	Входные данные	Выходные данные
1	5	<b>15</b>	665857
2	5	<b>3</b>	17
3	5	<b>1</b>	3
4	5	<b>10</b>	8119
5	5	<b>8</b>	1393

### Задача 6.

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
<b>AB 2</b>	<b>4 12</b>

#### *Пояснение:*

Получится составить следующие 6 слов максимальной длины:

**AB ABV AAV 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;
```

```
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;
```

```
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