

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации
«Профессор Лебедев» (общеобразовательный предмет информатика), весна 2019 г.
10 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	1	1000 1000 1000	1

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 к 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.

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

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 *01*. Иначе число не изменяется. Например,

19_{10}

00010011_2

Преобразуется в

$__000100$ (сдвиг вправо)

01000100_2 (результат)

68_{10}

Напишите программу, которая вычисляет наибольшее число, которое можно получить при преобразовании чисел из интервала $[A; B]$.

На вход программы подаётся два целых числа A и B , записанные через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Пример

Ввод	Вывод
18 20	68

```
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
```

```

else
  f:=64+ n div 4
end;

var a,b,i,x,max,imax:integer;
begin
  readln(a,b) ;
  for i:=a to b do
    begin
      x:=f(i);
      if (i=a)or (max<x) then begin
        max:=x;
        imax:=i
      end;
    end;
  writeln(max);
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	3	1 10	66
2	3	1 63	79
3	3	64 72	81
4	3	145 146	100
5	3	240 254	252

Задача 4.

Прототип робота-уборщика выполняет 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 5.

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
<i>AB</i>	<i>3 9</i>

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 VAB VBA**

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	6	ABDC	259808
5	8	ABCDE	22288

Задача 6.

Вася пытается вспомнить пароль от архива с конспектами. К сожалению, он помнит только то, что в пароле не очень много символов, и он состоял из букв *A,a,o,U,y,b,v,z,d*. Еще он точно

уверен, что если читать пароль с начала, то количество гласных букв в последовательности уже прочитанных знаков никогда не превысит количество согласных. Например, комбинация *бво* может быть паролем, а вот комбинация *абв* – нет. Помогите Васе оценить масштаб проблемы и напишите программу, которая будет рассчитывать количество возможных паролей заданной длины.

Формат ввода

В строке вводится целое положительное число n – количество символов в пароле ($n \leq 10$).

Формат вывода

Вывести одно целое число – количество способов, которыми можно собрать пароль заданной длины, удовлетворяющий наложенным условиям. Если не существует ни одной такой комбинации, то вывести 0 .

Пример

Входные данные	Выходные данные
2	36

Решение

```

program my;
const N=10;
      M=N;

type tmatrix = array [0..N,0..M] of longint;

procedure init(var matrix:tmatrix);
var i,j:integer;
begin
  for i:=0 to N do
    for j:=0 to M do
      matrix[i,j]:=0;
    matrix[0,0]:=1;
  end;

function calc(matrix:tmatrix; kol:integer):longint;
var i,j:integer;
begin
  for i:=1 to kol do
    begin
      for j:=0 to M-1 do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j+1]*5;

      for j:=1 to M do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j-1]*4;
    end;

```



```

j:=0;
for i:=0 to M do
  j:=j+matrix[kol,i];
  calc:=j;
end;

var
  matrix:tmatrix;
  kol:integer;
begin
  readln(kol);
  init(matrix);
  writeln(calc(matrix,kol));
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	5	3	224
2	5	5	14144
3	3	1	4
4	7	6	127296
5	10	8	8510976

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации
«Профессор Лебедев» (общеобразовательный предмет информатика), весна 2019 г.
10 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	1	23 50 4 3 20	0 0

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.

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

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. Иначе число не изменяется. Например,

19_{10}

00010011_2

Преобразуется в

$010011_ _$ (сдвиг влево)

01001101_2 (результат)

77_{10}

Напишите программу, которая вычисляет наибольшее число, которое можно получить при преобразовании чисел из интервала $[A; B]$.

На вход программы подаётся два целых числа A и B , записанные через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Пример

Ввод	Вывод
18 20	77

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
else
f:=1+ (n * 4)mod 256
end;

var a,b,i,x,max,imax:integer;
begin
readln(a,b) ;
for i:=a to b do
begin
x:=f(i);
if (i=a)or (max<x) then begin
max:=x;
imax:=i
end;
end;
writeln(max);
end.
```

Тесты

N	Оценка	Входные данные	Выходные данные
1	3	17 18	18
2	3	20 26	105
3	3	30 36	141
4	3	161 162	137
5	3	125 130	253

Задача 4.

Захват манипулятора способен выполнить три команды: «взять», «держат» 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	15	803759
2	5	3	19

3	5	<i>I</i>	2
4	5	<i>IO</i>	9799
5	5	<i>8</i>	1680

Задача 5.

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
<i>AB 2</i>	<i>6 24</i>

Пояснение:

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

BAVA BVAA

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

N	Оценка	Входные данные	Выходные данные
1	3	AB 5	252 2520
2	5	ABC 2	90 540
3	5	ABCD 2	2520 20160
4	6	ABC 5	756756 11351340
5	8	ABCD 4	63063000 1009008000

Задача 6.

Коля хочет защитить файл от копирования и вводит пароль. Он принял решение использовать только десятичные цифры **0,1,2,3,4,5** и накладывает дополнительное условие. При посимвольном прочтении пароля количество четных цифр в последовательности уже прочитанных знаков никогда не превысит количество нечетных. При этом Коля почему-то не считает ноль четной цифрой, и поэтому просто игнорирует его при подсчете. Например, комбинация **012354** может быть паролем, а вот комбинация **211** – нет. Помогите Коле проверить надежность пароля и напишите программу, которая будет рассчитывать количество возможных паролей заданной длины.

Формат ввода

В строке вводится целое положительное число n – количество символов в пароле ($n \leq 10$).

Формат вывода

Вывести одно целое число – количество способов, которыми можно собрать пароль заданной длины, удовлетворяющий наложенным условиям. Если не существует ни одной такой комбинации, то вывести **0**.

Пример

Входные данные	Выходные данные
2	22

Решение

```
program my;
const N=10;
      M=N;

type tmatrix = array [0..N,0..M] of longint;

procedure init(var matrix:tmatrix);
var i,j:integer;
begin
  for i:=0 to N do
```

```

for j:=0 to M do
  matrix[i,j]:=0;
matrix[0,0]:=1;
end;

function calc(matrix:tmatrix; kol:integer):longint;
var i,j:integer;
begin
  for i:=1 to kol do
    begin
      for j:=0 to M-1 do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j+1]*2;
      for j:=0 to M do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j];
      for j:=1 to M do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j-1]*3;
      end;
    j:=0;
    for i:=0 to M do
      j:=j+matrix[kol,i];
    calc:=j;
  end;

var
  matrix:tmatrix;
  kol:integer;
begin
  readln(kol);
  init(matrix);
  writeln(calc(matrix,kol));
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	5	3	118
2	5	5	3802
3	3	1	4
4	7	6	21970
5	10	8	743494

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

Вариант 3

Задача 1.

Андрей обожает рельсовый транспорт. В кармане он всегда носит свой любимый трамвай *Tatpa T3*, 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_2 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	1	0 0 1 1 1 0	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;
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.

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

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. Иначе число не изменяется. Например,

12_{10}

00001100_2

Преобразуется в

$001100_ _$ (сдвиг влево)

00110010_2 (результат)

50_{10}

Напишите программу, которая вычисляет наибольшее число, которое можно получить при преобразовании чисел из интервала $[A; B]$.

На вход программы подаётся два целых числа A и B , записанные через пробел.

Программа должна вывести одно целое число - вычисленное значение.

Пример

Ввод	Вывод
11 13	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:=n
else
f:=2+ (n * 4)mod 256
end;

var a,b,i,x,max,imax:integer;
begin
readln(a,b) ;
for i:=a to b do
begin
x:=f(i);
if (i=a)or (max<x) then begin
max:=x;
imax:=i
end;
end;
writeln(max);
end.
```

Тесты

N	Оценка	Входные данные	Выходные данные
1	3	87 88	88
2	3	20 26	98
3	3	31 37	146
4	3	165 166	154
5	3	60 70	254

Задача 4.

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	<i>15</i>	665857
2	5	<i>3</i>	17
3	5	<i>1</i>	3
4	5	<i>10</i>	8119
5	5	<i>8</i>	1393

Задача 5.

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
<i>AB 2</i>	<i>4 12</i>

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;
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	6	Abcdefg 10	10000000 385000000
5	8	Abcdefghij 6	60466176 2116316160

Задача 6.

Петя пытается подобрать надежный и легко запоминающийся пароль. Он принял решение использовать только десятичные цифры **0,1,2,3,4,5** и накладывает дополнительное условие. При посимвольном прочтении пароля количество цифр **1** в последовательности уже прочитанных знаков никогда не превысит количество цифр **0**. Например, комбинация **012354** может быть паролем, а вот комбинация **2110** – нет. Помогите Петя проверить надежность пароля и напишите программу, которая будет рассчитывать количество возможных паролей заданной длины.

Формат ввода

В строке вводится целое положительное число **n** – количество символов в пароле ($n \leq 10$).

Формат вывода

Вывести одно целое число – количество способов, которыми можно собрать пароль заданной длины, удовлетворяющий наложенным условиям. Если не существует ни одной такой комбинации, то вывести **0**.

Пример

Входные данные	Выходные данные
2	26

Решение

```

program my;
const N=10;
M=N;
type tmatrix = array [0..N,0..M] of longint;
procedure init(var matrix:tmatrix);
var i,j:integer;

```

```

begin
  for i:=0 to N do
    for j:=0 to M do
      matrix[i,j]:=0;
      matrix[0,0]:=1;
    end;
  end;

function calc(matrix:tmatrix; kol:integer):longint;
var i,j:integer;
begin
  for i:=1 to kol do
    begin
      for j:=0 to M-1 do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j+1];
      for j:=0 to M do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j]*4;
      for j:=1 to M do
        matrix[i,j]:=matrix[i,j]+matrix[i-1,j-1];
      end;
      j:=0;
    end;
  for i:=0 to M do
    j:=j+matrix[kol,i];
  calc:=j;
end;

var
  matrix:tmatrix;
  kol:integer;
begin
  readln(kol);
  init(matrix);
  writeln(calc(matrix,kol));
end.

```

Тесты

N	Оценка	Входные данные	Выходные данные
1	5	3	139
2	5	5	4194
3	3	1	5
4	7	6	23460
5	10	8	751526

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело»
специализации «Профессор Лебедев», весна 2019 г.
10-11 класс**

Ситуационная задача

Вариант – 1

Сердечник подкалиберного снаряда представляет из себя комбинацию тел вращения, образованных различными кривыми. Общий вид уравнения кривой выглядит как

$$y = a \cdot (x + b) \cdot x + d \cdot \ln(x + c).$$

Зная количество тел вращения, уравнения кривых и границы отрезков, на которых эти кривые применяются, найдите массу сердечника. Сердечник сделан из обеднённого урана.

Формат ввода

В строке вводится натуральное число N – число тел вращения. N не превышает 5.

Далее следует N шестёрок вещественных неотрицательных чисел x_1, x_2, a, b, c, d – соответственно, координаты границ отрезка, и коэффициенты при кривой на этом отрезке. Все числа отделены друг от друга одним или несколькими пробелами.

Гарантируется, что разрывов нет и каждый следующий отрезок начинается там, где закончился предыдущий.

Гарантируется, что каждое уравнение имеет математический смысл.

Никакие числа не превышают **1000**.

Формат вывода

На выходе программа должна выдать целое число – массу сердечника в граммах, округлённую до ближайшего целого.

Плотность обеднённого урана – **19,05 г/см³**. Все значения переменных даны в сантиметрах.

Примечание: число требуется вывести с точностью до целого только ради исключения накладок при тестировании. Рекомендуется считать суммы с заранее известной точностью $\varepsilon = 10^{-5}$. Точность ε считать достигнутой, когда при вычислении интегральной суммы уменьшение отрезка x вдвое приводит к изменению суммы меньше, чем на ε

Пример

Входные данные	Выходные данные
1 0 1 1 1 1 1	125

Решение

```
program z17;
const
  ro=19.05;
  pi=3.1415926;
  eps = 0.00001;

function f(x,a,b,c,d:real):real;
begin
  f:=a*x*(x+b)+d*ln(x+c);
end;

function work(x0,x1,a,b,c,d:real):real;
var vt,pv,x,h:real;
begin
  vt:=sqr(f((x1+x0)/2,a,b,c,d))*(x1-x0);
  h:=(x1-x0)/2;
  repeat
    pv:=vt;
    x:=x0;
    vt:=0;
    while x<x1 do
      begin
        vt:=vt+sqr(f(x,a,b,c,d))*h;
        x:=x+h;
      end;
    h:=h/2;
  until abs(vt-pv)<eps;
  work:=ro*pi*vt;
end;

var w,x0,x1,a,b,c,d:real;
    i,n:integer;
begin
  w:=0;
  read(n);
  for i:=1 to n do
    begin
      read(x0);
      read(x1);
      read(a);
      read(b);
      read(c);
      read(d);
      w:=w+work(x0,x1,a,b,c,d);
    end;
  writeln(round(w));
end.
```

Тесты

N	Оценка	Входные данные	Выходные данные
1	10	1 0 1 1 0 1 0	12
2	10	2 0 1 1 1 1 1 2 1 2 1 3	4207
3	10	3 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1	375

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело»
специализации «Профессор Лебедев», весна 2019 г.
10-11 класс**

Ситуационная задача

Вариант – 2

Сердечник подкалиберного снаряда представляет из себя комбинацию тел вращения, образованных различными кривыми. Общий вид уравнения кривой выглядит как

$$y = a \cdot x^2 + 2 \cdot (b+c) \cdot x + d \cdot \ln(x+4).$$

Зная количество тел вращения, уравнения кривых и границы отрезков, на которых эти кривые применяются, найдите массу сердечника. Сердечник сделан из обеднённого урана.

Формат ввода

В строке вводится натуральное число N – число тел вращения. N не превышает 5.

Далее следует N шестёрок вещественных неотрицательных чисел x_1, x_2, a, b, c, d – соответственно, координаты границ отрезка, и коэффициенты при кривой на этом отрезке. Все числа отделены друг от друга одним или несколькими пробелами.

Гарантируется, что разрывов нет и каждый следующий отрезок начинается там, где закончился предыдущий.

Гарантируется, что каждое уравнение имеет математический смысл.

Никакие числа не превышают **1000**.

Формат вывода

На выходе программа должна выдать целое число – массу сердечника в граммах, округлённую до ближайшего целого.

Плотность обеднённого урана – **19,05 г/см³**. Все значения переменных даны в сантиметрах.

Примечание: число требуется вывести с точностью до целого только ради исключения накладок при тестировании. Рекомендуется считать суммы с заранее известной точностью $\varepsilon = 10^{-5}$. Точность ε считать достигнутой, когда при вычислении интегральной суммы уменьшение отрезка x вдвое приводит к изменению суммы меньше, чем на ε

Пример

Входные данные	Выходные данные
1 0 1 1 1 1 1	1017

Решение

```
program z17;
const
  ro=19.05;
  pi=3.1415926;
  eps = 0.00001;

function f(x,a,b,c,d:real):real;
begin
  f:=a*x*x+2*(b+c)*x+d*ln(x+4);
end;

function work(x0,x1,a,b,c,d:real):real;
var vt,pv,x,h:real;
begin
  vt:=sqr(f((x1+x0)/2,a,b,c,d))*(x1-x0);
  h:=(x1-x0)/2;
  repeat
    pv:=vt;
    x:=x0;
    vt:=0;
  while x<x1 do
    begin
      vt:=vt+sqr(f(x,a,b,c,d))*h;
      x:=x+h;
    end;
    h:=h/2;
  until abs(vt-pv)<eps;
  work:=ro*pi*vt;
end;

var w,x0,x1,a,b,c,d:real;
    i,n:integer;
begin
  w:=0;
  read(n);
  for i:=1 to n do
    begin
      read(x0);
      read(x1);
      read(a);
      read(b);
      read(c);
      read(d);
      w:=w+work(x0,x1,a,b,c,d);
    end;
  writeln(round(w));
end.
```


Тесты

N	Оценка	Входные данные	Выходные данные
1	10	1 0 1 1 0 0 0	12
2	10	2 0 1 1 1 1 1 2 1 2 1 1	11609
3	10	3 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1	3050

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело»
специализации «Профессор Лебедев», весна 2019 г.
10-11 класс**

Ситуационная задача

Вариант – 3

Сердечник подкалиберного снаряда представляет из себя комбинацию тел вращения, образованных различными кривыми. Общий вид уравнения кривой выглядит как

$$y = a \cdot x^{1/2} + 2 \cdot (b \cdot x^{1/2} + c) \cdot x + d \cdot \ln(x+1).$$

Зная количество тел вращения, уравнения кривых и границы отрезков, на которых эти кривые применяются, найдите массу сердечника. Сердечник сделан из обеднённого урана.

Формат ввода

В строке вводится натуральное число N – число тел вращения. N не превышает 5.

Далее следует N шестёрок вещественных неотрицательных чисел x_1, x_2, a, b, c, d – соответственно, координаты границ отрезка, и коэффициенты при кривой на этом отрезке. Все числа отделены друг от друга одним или несколькими пробелами.

Гарантируется, что разрывов нет и каждый следующий отрезок начинается там, где закончился предыдущий.

Гарантируется, что каждое уравнение имеет математический смысл.

Никакие числа не превышают **1000**.

Формат вывода

На выходе программа должна выдать целое число – массу сердечника в граммах, округленную до ближайшего целого.

Плотность обеднённого урана – **19,05 г/см³**. Все значения переменных даны в сантиметрах.

Примечание: число требуется вывести с точностью до целого только ради исключения накладок при тестировании. Рекомендуется считать суммы с заранее известной точностью $\varepsilon = 10^{-5}$. Точность ε считать достигнутой, когда при вычислении интегральной суммы уменьшение отрезка x вдвое приводит к изменению суммы меньше, чем на ε

Пример

Входные данные	Выходные данные
1 0 1 1 1 1 1	640

Решение

```
program z17;
const
  ro=19.05;
  pi=3.1415926;
  eps = 0.00001;

function f(x,a,b,c,d:real):real;
begin
  f:=a*sqrt(x)+2*(b*sqrt(x)+c)*x+d*ln(x+1);
end;

function work(x0,x1,a,b,c,d:real):real;
var vt,pv,x,h:real;
begin
  vt:=sqr(f((x1+x0)/2,a,b,c,d))*(x1-x0);
  h:=(x1-x0)/2;
  repeat
    pv:=vt;
    x:=x0;
    vt:=0;
    while x<x1 do
      begin
        vt:=vt+sqr(f(x,a,b,c,d))*h;
        x:=x+h;
      end;
    h:=h/2;
  until abs(vt-pv)<eps;
  work:=ro*pi*vt;
end;

var w,x0,x1,a,b,c,d:real;
    i,n:integer;
begin
  w:=0;
  read(n);
  for i:=1 to n do
    begin
      read(x0);
      read(x1);
      read(a);
      read(b);
      read(c);
      read(d);
      w:=w+work(x0,x1,a,b,c,d);
    end;
  writeln(round(w));
end.
```

Тесты

N	Оценка	Входные данные	Выходные данные
1	10	1 0 1 1 0 0 0	30
2	10	2 0 1 1 1 1 1 1 2 1 2 1 1	10624
3	10	3 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1	1921