

Написать программу, которая считывает пароль. Если пароль неверен, то сообщает об этом и просит ввести пароль заново, иначе выводится сообщение, что пароль верный.

```
Program parol; //программа на проверку пароля
var
pas1:string;
const
pas2='шКола';
begin
repeat
  write('введите пароль: ');
  readln(pas1);
  if pas1<>pas2 then write('Ошибка! Пароль неверен! Заново ');
until pas1=pas2;
writeln('Пароль верен, программа выполняется дальше');
end.
```

Вложенные ЦИКЛЫ

Для заданного натурального числа n требуется найти все такие тройки натуральных чисел a, b, c , что $a+b+c=n$.

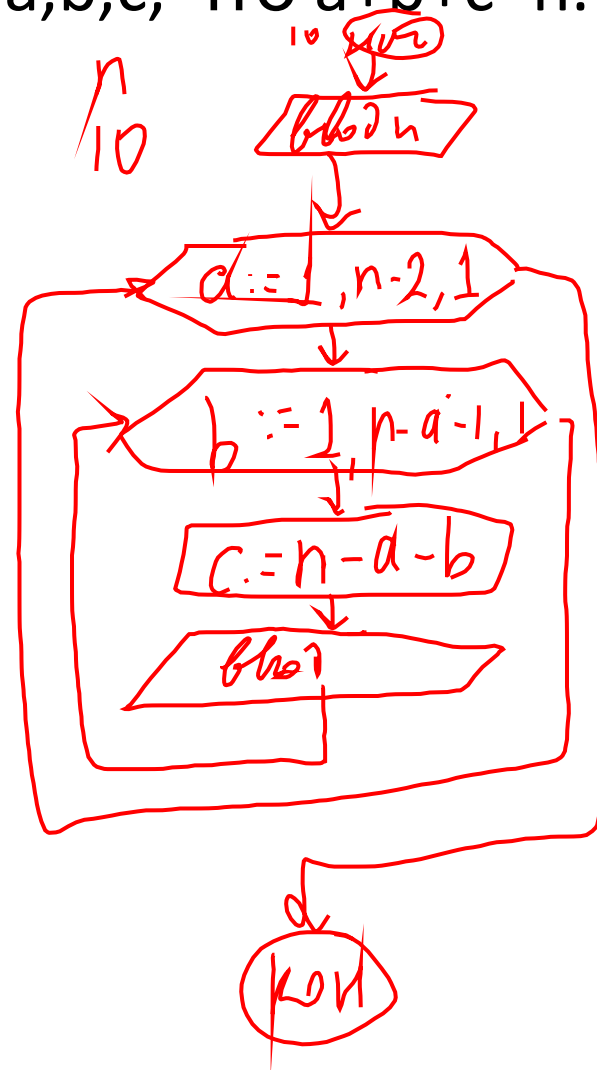
```

for a := 1 to n-2 do
  for b := 1 to n-a-1 do
    begin
      c := n-a-b;
      writeln(a, '+', b, '+', c, '=', n)
    end
  end
end

```

n
10

a	b	c
1	1	8
1	2	7
1	8	1
2	1	7
2	2	6
2	8	1
3		



Требуется распечатать все трёхзначные числа, в которых есть две одинаковые цифры.

```
for i:=1 to 9 do
  for j:=0 to 9 do
    for k:=0 to 9 do
      if (i=j) or (j=k) or (i=k) then writeln(i,j,k);
```

Если порядок не важен:

$i=2$ $j=$

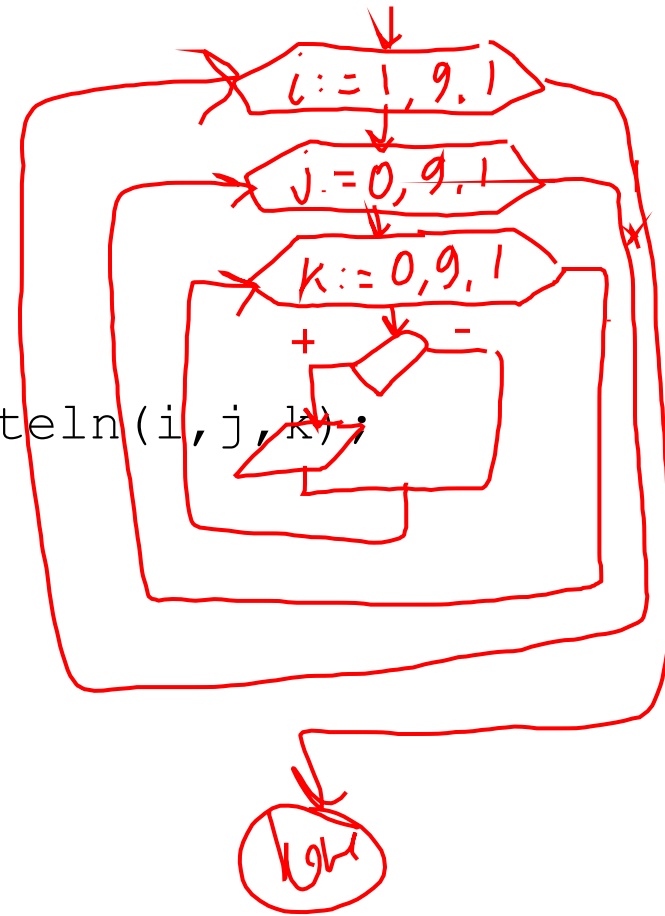
Handwritten examples of three-digit numbers with two identical digits:

6 3 6
3 6 6
6 3 3

```
for i:=1 to 9 do
  begin
    writeln(i,0,0, ' ', i,i,0, ' ', i,0,i);
    for j:=1 to 9 do
      writeln(j,j,i, ' ', i,j,j, ' ', j,i,j)
    end
```

Handwritten examples of three-digit numbers with two identical digits:

100 110 101
111 111 111
321 122 212



В приведенном решении удалось избежать использования условных операторов.

Два последних примера показывают, что в задачах по информатике, для того чтобы получить ответ, не всегда нужно оперировать именно теми объектами, о которых идет речь в условии. В этих задачах совершенно излишним было бы перебирать все трехзначные числа, выделять из них цифры и сравнивать их между собой. Более того, формировать из подходящих цифр число, только для того чтобы его распечатать, тоже не нужно: печать нескольких цифр подряд без разделителей приведет к тому, что зрительно эти цифры как раз и образуют нужное число.

Для заданного натурального числа n найти все тройки натуральных чисел a, b, c таких, что $a + b + c = n$ и $a \leq b \leq c$. Минимизируйте количество условий, которые используются в программе.

$$\begin{array}{l} 1+1+3=5 \\ 1+2+2=5 \\ -1+3+1=5 \\ -2+1+2=5 \\ \quad \underline{a} \quad \underline{b} \end{array}$$

Входные данные

На вход программе подается натуральное число $n \leq 100\,000$.

Выходные данные

Выведите количество искоемых троек.

Примеры

входные данные
3
выходные данные
1

входные данные
2
выходные данные
0

```
var
a,b,c,n:longint;
kol:integer;
begin
  readln(n);
  for a:= 1 to n-2 do
    for b:= a to n-a-1 do
      begin
        c:=n-a-b;
        if c>=b then kol:=kol+1;
      end;
    end;
  writeln(kol);
end.
```