

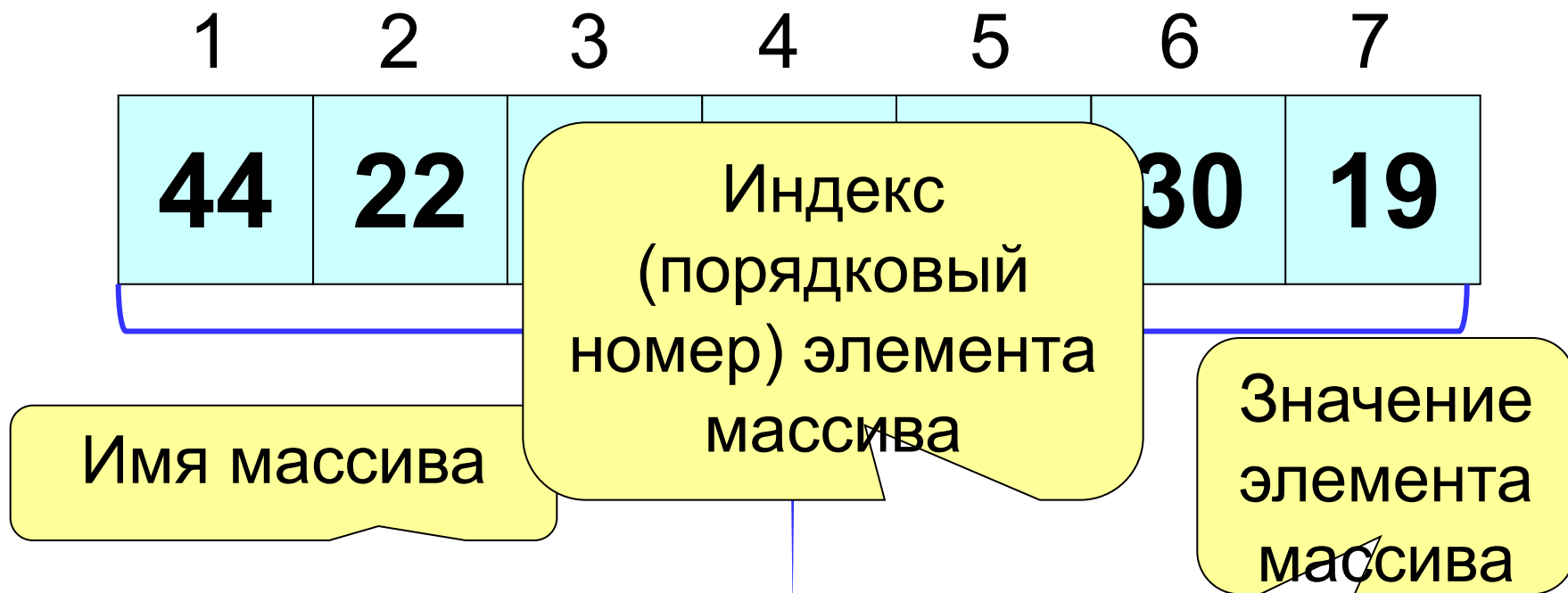
# Одномерные массивы

Презентация по программированию

Автор: учитель информатики  
МОУ Плесской СОШ  
Юдин А.Б.

2012 год

**Массив** – пронумерованная совокупность ячеек памяти, названная одним именем



```
A [4] := 11;
```

Объявление массива в разделе описания переменных:

**Var имя:ARRAY[1..N] OF тип;**

Имя массива

Количество  
элементов

Тип элементов

**Var A:Array[1..10] of Integer;**

**Var Tem:Array[1..100] of Real;**

```
Const  
m=10;
```

Количество элементов можно описать в разделе констант

```
Var a : array [1.. m] of integer;
```

Иногда массив объявляют как пользовательский тип данных

```
Type t=array[1..10] of integer;  
Var a : t;
```

**Const** a:array[1..5] of integer =  
(3,-2,1,4,3);

Массив констант

**Var** R:Array [-20..20] of Real;  
**Var** N:Array ['A'..'Z'] of Integer;

В зависимости от задачи индексы элементов могут начинаться не только с единицы или быть символьного типа

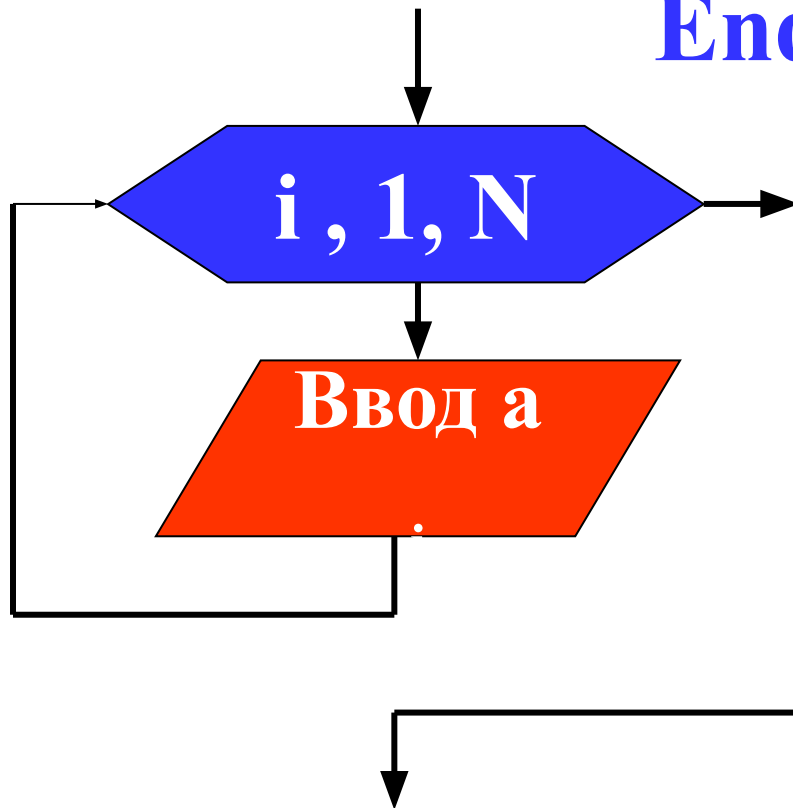
# 1. Заполнение массива с клавиатуры

**For i:=1 to N do begin**

**Write(' a [ ', i, ' ] = ' );**

**ReadLn ( a [ i ] );**

**End;**



## 2. Заполнение массива случайными числами.

**For i:=1 to N do begin**

**a[i]:=Random(B-A)+A;**

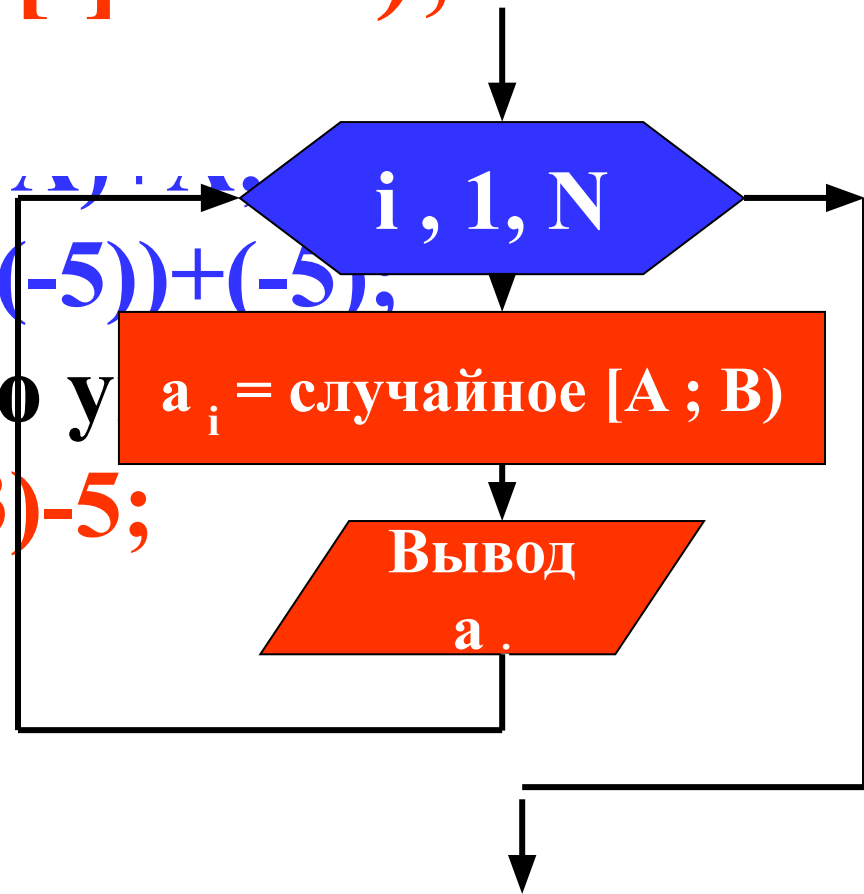
**WriteLn(a[i]:10:5);**

**End;**

**m[i] := RANDOM(8-(-5))+(-5);**

такую запись нужно у

**m[i] := RANDOM(13)-5;**



### 3. Заполнение массива при помощи прогрессии.

**$a[1] := 3;$**

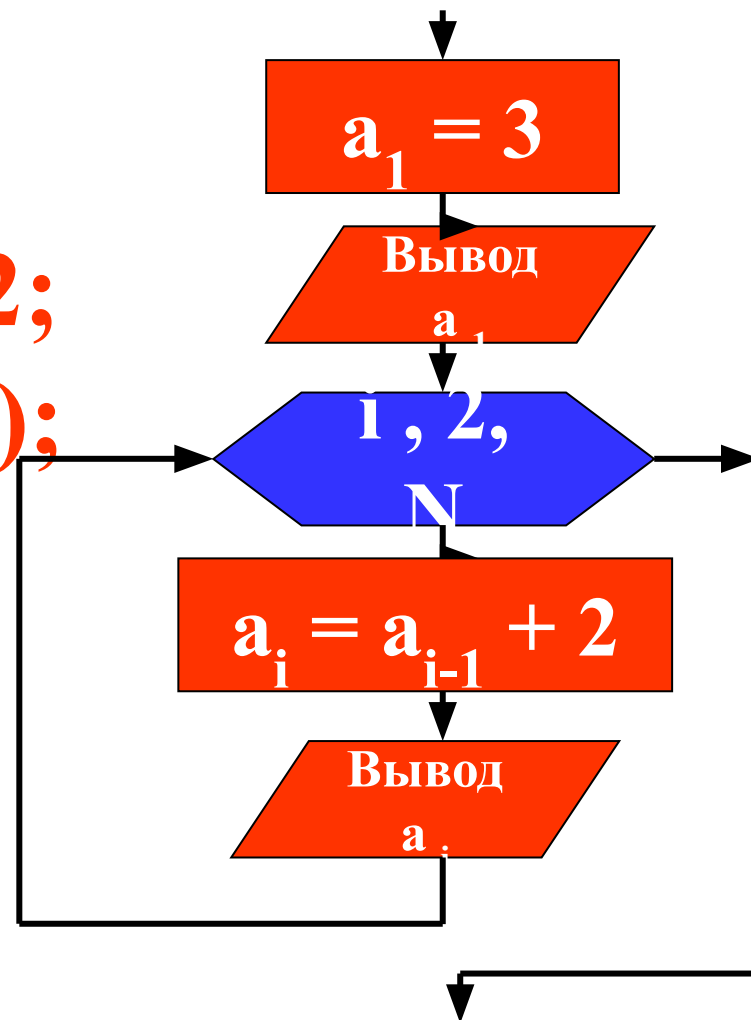
**Writeln( $a[1]$ );**

**For  $i := 2$  to  $N$  do begin**

**$a[i] := a[i-1] + 2;$**

**Writeln( $a[i]:5$ );**

**End;**





**а) в столбик:**

**For i:=1 to N do WriteLn(a[i]:4:2);**

**б) в строку**

**For i:=1 to N do Write(a[i]:4:2);**

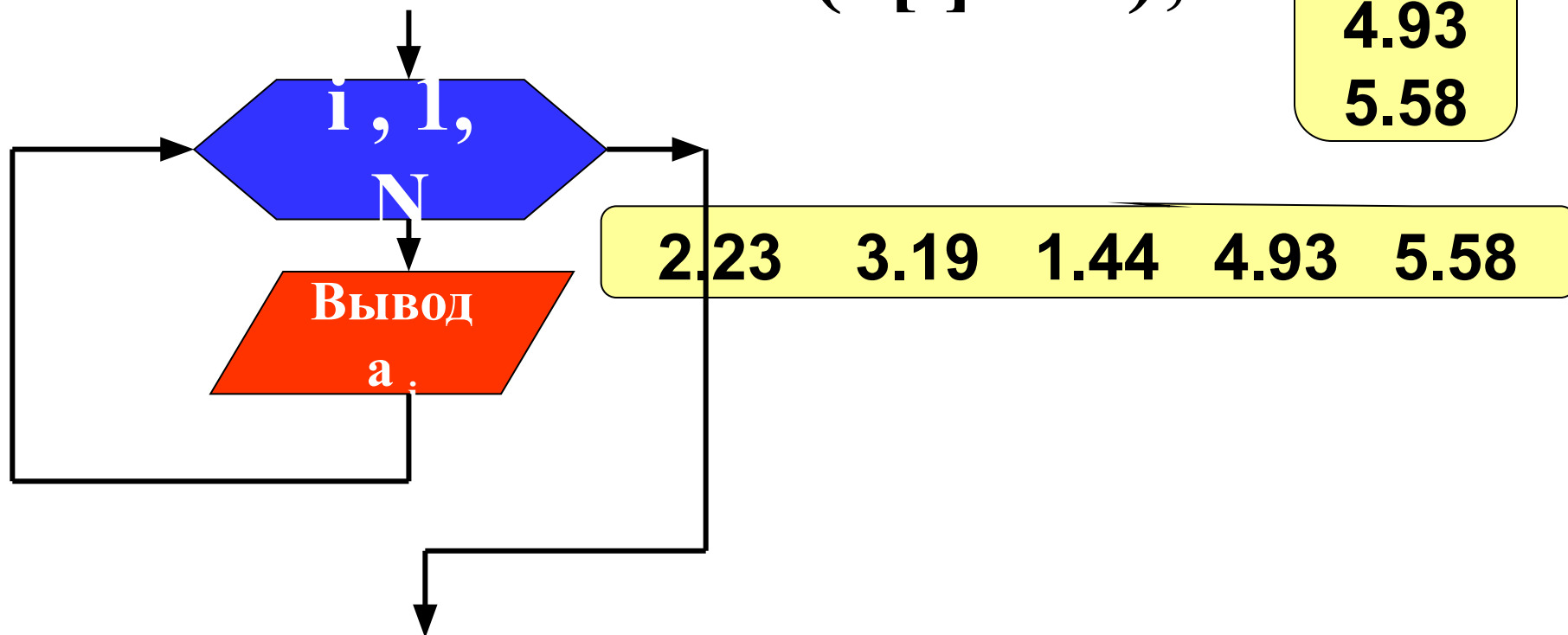
2.23

3.19

1.44

4.93

5.58



2.23 3.19 1.44 4.93 5.58

В: CRT - программа завершена

-10	-5	-5	-4	-9	5	-6	-5	3	9
-9	-4	-4	-3	-8	6	-5	-4	4	10

первая строка закончена, переход на вторую

```
a[i]:= random(21)-10;
```

```
write(a[i]:5);
```

```
end;
```

```
writeln;
```

```
for i:=1 to 10 do begin
```

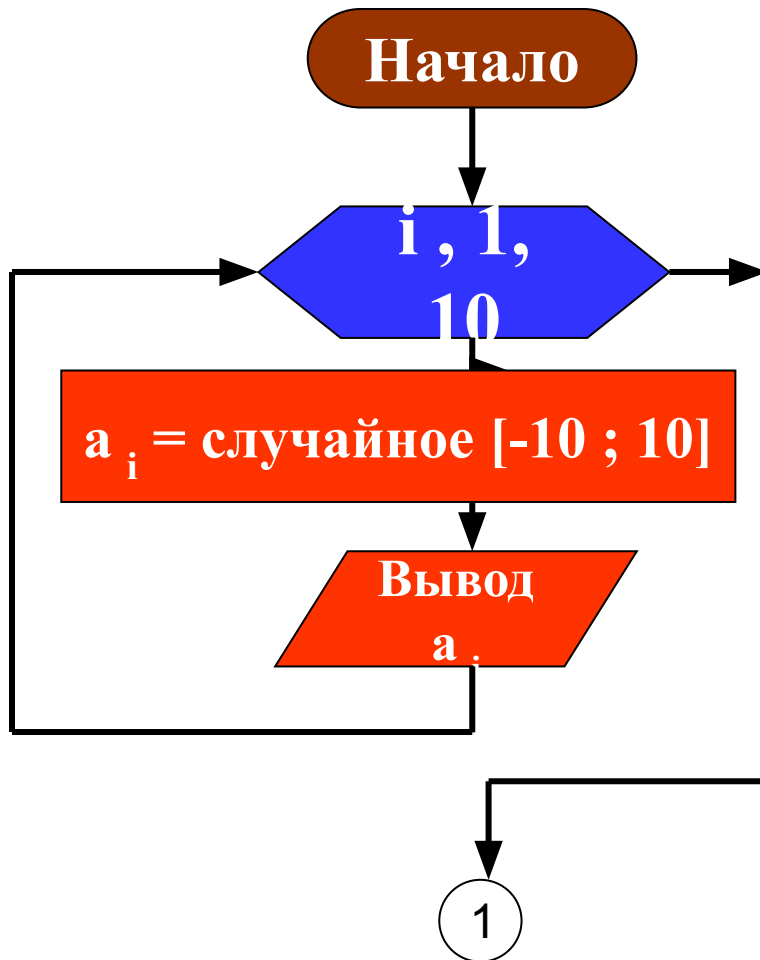
```
  a[i]:=a[i]+1;
```

```
  write(a[i]:5);
```

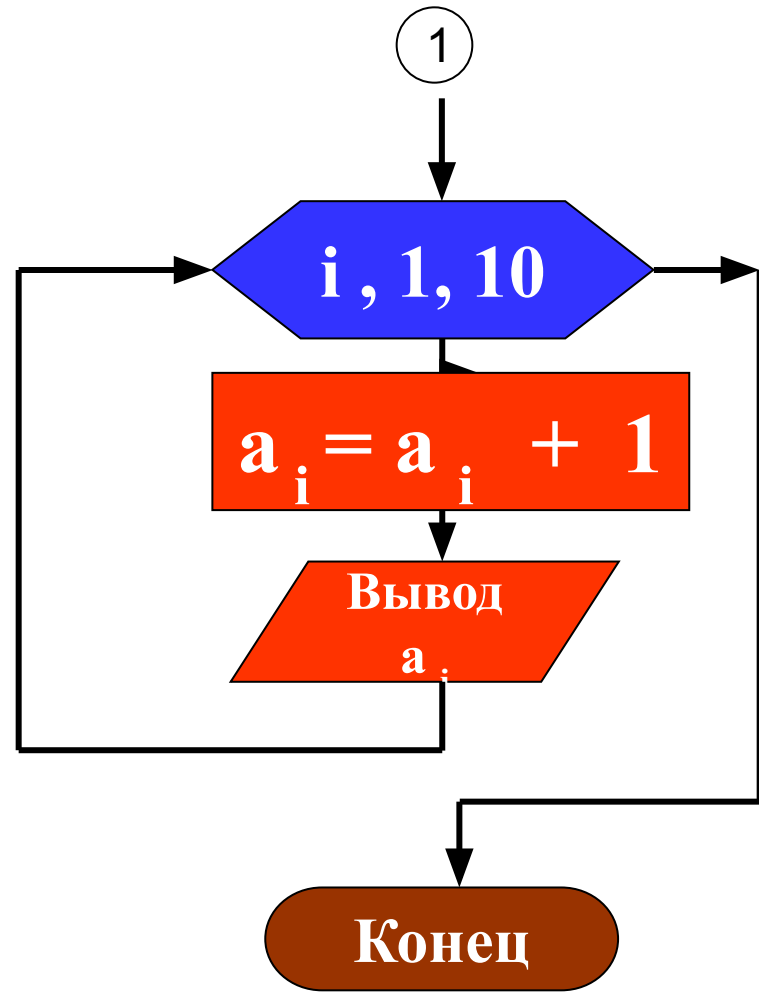
```
end;
```

```
end.
```

Оператор вывода без LN, вывод осуществляется в строку



**Пустой Writeln между циклами обычно в блок-схеме не изображают.**



**Разрыв блок-схемы, если она целиком на лист не убирается**

**Prog**

CRT - программа завершена

**Uses**

1 4 9 16 25 36 49 64 81 100

**var a****i:integer;****begin****clrscr;****for i:=1 to 10 do begin****a[i]:=i\*i;****write(a[i]:5);****end;****end.**

Значение элемента массива  
зависит только от  
порядкового номера

Uses Crt:

```
var a:array
```

```
  i:integer
```

```
begin
```

```
  clrscr;
```

```
  a[1]:=2;
```

```
  write(a[1]:5);
```

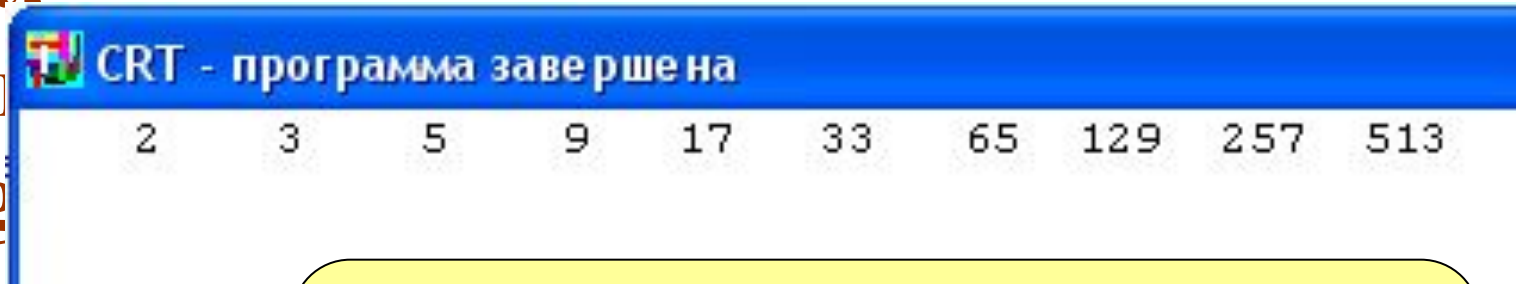
```
  for i:=2 to 10 do begin
```

```
    a[i]:=2*a[i-1]-1;
```

```
    write(a[i]:5);
```

```
  end;
```

```
end.
```



Значение элемента массива  
зависит от предшествующего  
элемента

## Изменение элементов массива удовлетворяющих условию

**For i:=1 to N do If (условие) then a[i]:=значение:**

### Изменения:

**a [ i ] :=5** – заменить i-ый элемент на число 5

**a [ i ] := - a [ i ]** – заменить i-ый элемент на противоположный по знаку

**a [ i ] := a [ i ] \* 2** – удвоить каждый элемент массива

**a [ i ] := a [ i - 1 ] + 3** –заменить i-ый элемент предшествующим плюс 3

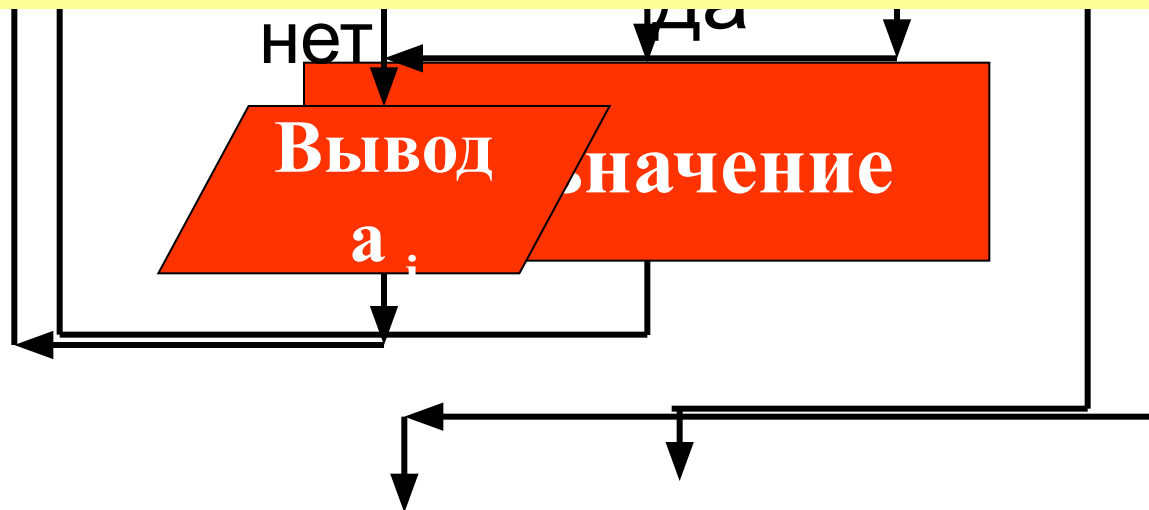
# Цикл обработки и вывода элементов массива на экран

**For** i:=1 to N **do begin**

**If** (условие) **then** a[i]:=значение:

**Writeln**(a[i]:5);

**end;**



# Нахождение суммы элементов массива

Условие:  $\sum_{i=1}^N a_i = 0$

Предполагаем, что  
сумма равна нулю

**s:=0;**

**For i:=1 to N do If (условие) then s:=s+a[i];**

**WriteLn ('Сумма = ',**

Если элемент подходит  
по условию, добавляем

Нахождение

Предполагаем, что  
произведение равно

единице

, добавляем

**p:=1;**

его к сумме

**For i:=1 to N do If (условие) then p:=p\*a[i];**

**WriteLn ('Произведение = ', p:10:5);**



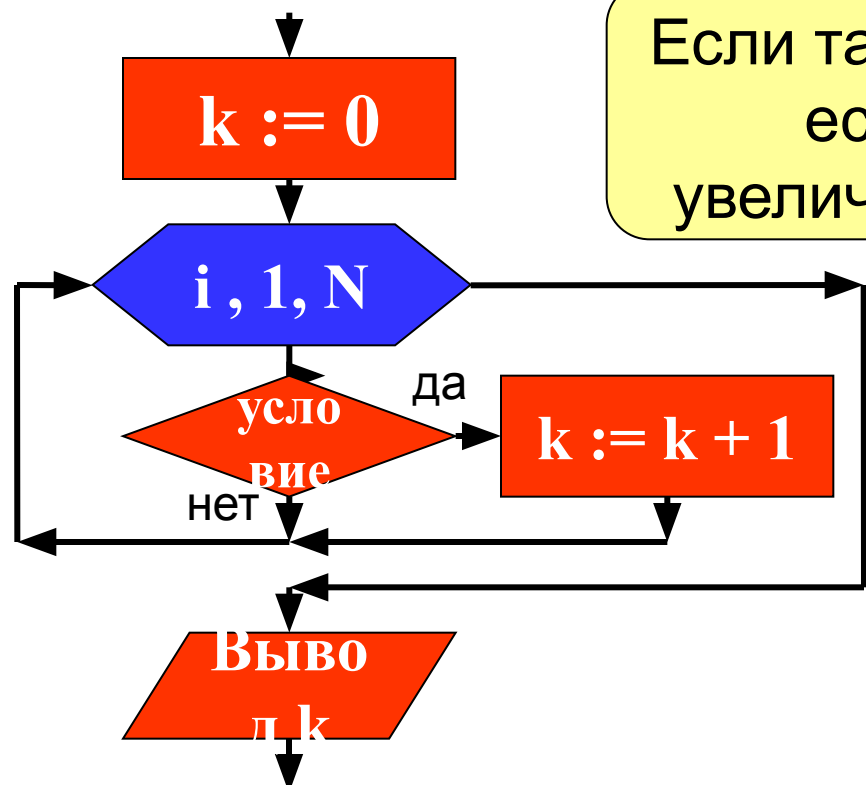
Предполагаем, что таких элементов нет

**элементов условию:**

**k:=0;**

**For i:=1 to N do If (условие) then k:=k+1;**  
**WriteLn('Кол-во=',k:10);**

Если такой элемент есть, то K увеличиваем на 1.



**Задача 4.** Дан массив из 15 целых чисел заполненный случайными числами из интервала [0; 20]. Выведите этот массив в строку и найдите количество четных элементов массива, которые меньше среднего арифметического его элементов.

```
ClrScr;
```

```
s:=0;
```

```
For i:=1 to 15 do begin
```

```
a[i]:=Random(21);
```

```
write (a[i]:10);
```

```
S:=S+a[i];
```

```
end;
```

Предполагаем  
сумма равна

Организуем цикл  
15 повт

Определяем число  
случайным образом

Добавляем  
к сумме

Выводим его на  
экран

**Writeln;**

Находим сред арифметичес

Выводим его на экран

**s:=s/15;**

Предполагаем, что таких э

Организуем цикл из 15 повторений

**Writeln('Сумма = ',s:10:5);**

**k:=0;**

**For i:=1 to 15 do**

**if (a[i] Mod 2 = 0) and (a[i]<s) then k:=k+1;**

**Writeln('Четных чисел меньших среднего арифметического ', k);**

Если элемент четный и меньше среднего арифметического, увеличиваем K на единицу

Выводим K на экран

# Нахождение максимального элемента в массиве и его номера:

```
max:=a[1];  
k:=1;
```

Предполагается, что  
наибольший элемент  
стоит на позиции 1.

Если находится  
большой элемент

```
for i:=2 to N do if (a[i]>max) then begin
```

Когда массив  
закончился, выводим  
наибольший элемент  
и его номер

Он становится  
максимальным

```
max:=a[i];  
k:=i;  
end;
```

```
WriteLn('Max=',max:10:5);  
WriteLn('Номер=',k:5);
```

## Иногда в задаче достаточно определить индекс

Предполагаем, что

элемент стоит  
в этом месте

CRT - программа завершена

```

max:=1;
for i:=1 to n do
  if a[i]>a[max] then
    max:=i;
writeln('Максимальный элемент ',a[max]);
writeln('Его номер  ',max);

```

`a[max]` then `max:=i;`

ый элемент ',a[max]);

Если  $i$ -ый  
стоящ

то, запоминаем его индекс в  
переменной max

# Сортировка элементов массива по

возра

Перебираем элементы  
от первого до  
предпоследнего

**For i:=1 to N-1 do**

**For j:=i to N do**

**if (a[i] > a[j]) then begin**

**p:=a[i];**

**a[i]:=a[j];**

**a[j]:=p;**

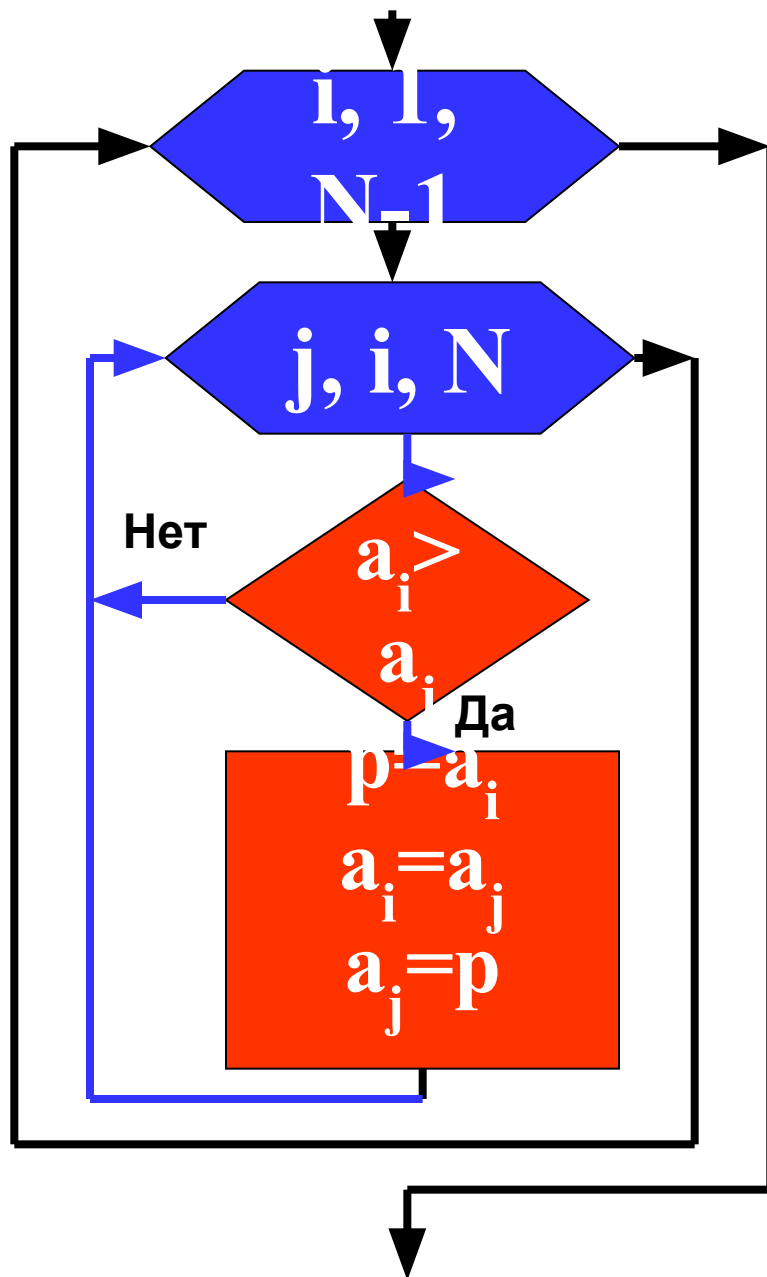
CRT - программа завершена

4	-2	6	-5
4	-2	6	-5
-2	4	6	-5
-2	4	6	-5
-5	4	6	-2
-5	4	6	-2
-5	4	6	-2
-5	-2	6	4
-5	-2	6	4
-5	-2	4	6
-5	-2	4	6

Сортировка массива:

4; -2; 6; -5

по возрастанию с  
изменениями  
происходящими в массиве



## Общая идея алгоритма:

1. Берем  $i$ -ый элемент
2. Последовательно сравниваем его со всеми элементами с права
3. Если находится элемент меньший чем  $i$ -ый, то они меняются местами
4. Так на  $i$ -ом месте окажется самый маленький элемент
5.  $i$  увеличиваем на единицу и повторяем процесс...

Упорядочить по возрастанию: 4 -2 6 -5

$i$	$a_i$	$j$	$a_j$	$a_i > a_j$

Массив			
4	-2	6	-5
-2	4	6	-5
-2	4	6	-5
-5	4	6	-2
-5	4	6	-2
-5	4	6	-2
-5	-2	6	4
-5	-2	6	4
-5	-2	4	6



## Вставка элемента с номером $p$ в последовательность

```
For  $i:=N$  downto  $p$  do  $a[i+1]:=a[i];$ 
```

```
 $a[p]:=$ значение;  
 $N:=N+1;$ 
```

Перебираем элементы от последнего до места

На место с индексом  $P$  ставим нужное значение и увеличиваем на 1 количество элементов

100

9

12

11

14

7

$p$

$N$

# Удаление элемента с номером $p$ из массива:

```
for i:=p to (N-1) do a[i]:=a[i+1];
```

```
N:=N-1;
```

Получаем элемент  
и уменьшаем на 1  
количество элементов

За каждый оборот  
цикла смещаем  
элементы на один  
номер в лево.

	100	9	12	11	14	7
--	-----	---	----	----	----	---

$p$

$N$

**Вставка элементов удовлетворяющих условию в другой массив. Пусть нам дан массив с элементами  $a[1..N]$ . Если элементы удовлетворяют условию  $P(x)$ , то, увеличиваем количество элементов в  $B$ , и под этим индексом помещаем элемент из  $A$ .**

**$j := 0;$**

**For  $i := 1$  to  $N$  do**

**if  $(a[i] \dots)$  then begin**

**$j := j + 1;$   
 $b[j] := a[i];$**

**$K := j;$**

Количество элементов в массиве  $B$  будет равно последнему порядковому номеру перенесенного элемента

Порядковый номер элемента в массиве  $B$

Если элемент удовлетворяет условию

То, увеличиваем количество элементов в  $B$ , и под этим индексом помещаем элемент из  $A$

// заполнить мас

r:= a[2]-a[1];

flag:=true;

For i:=2 to 9 do if (a[i+1]-a[i] < r) then flag:=false;

If flag then Begin

Writeln('Арифметическая');

Если значение истинно

End

else

Writeln('Не арифметическая');

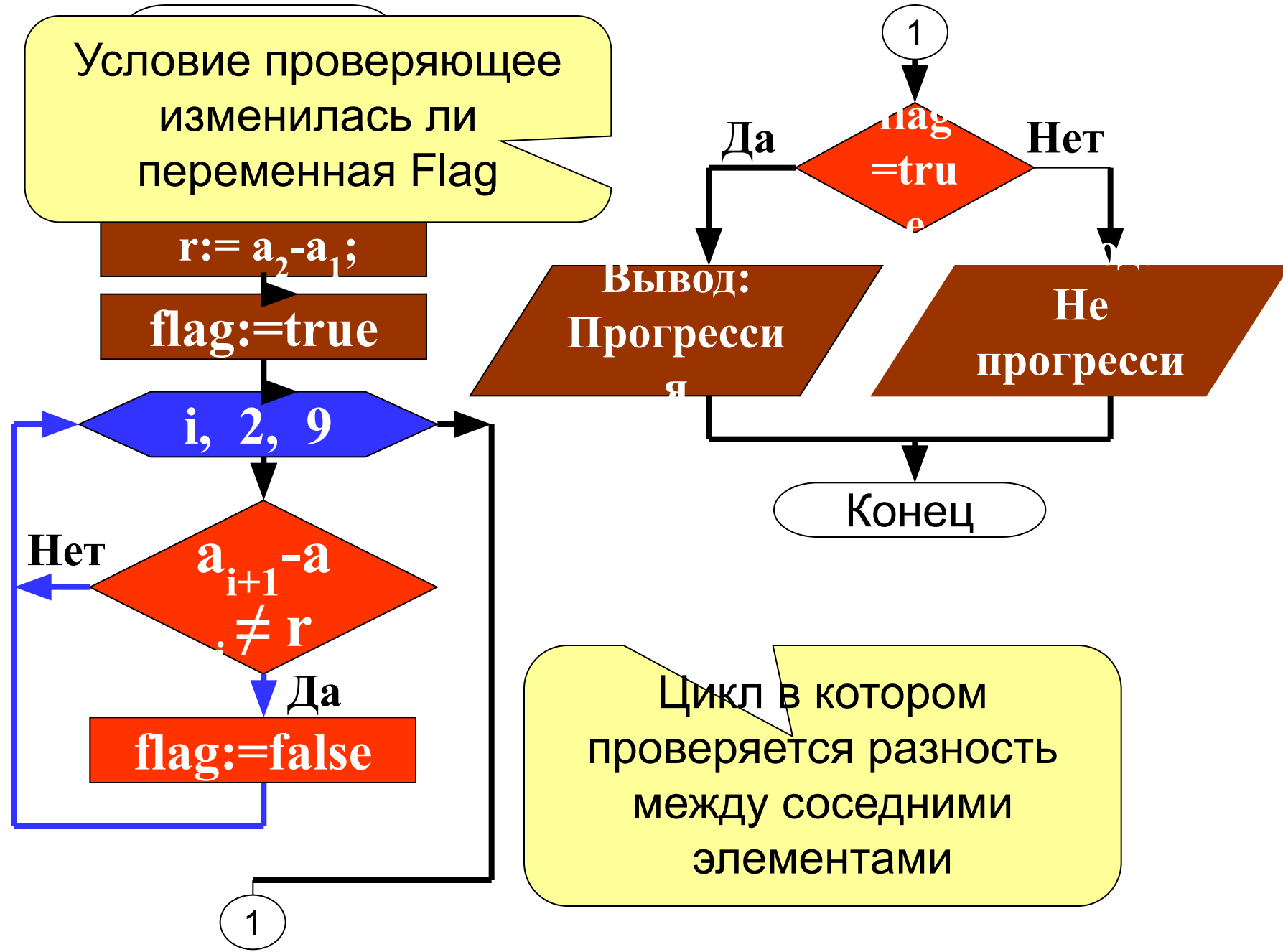
Находим разность

Организуем цикл от второго до предпоследнего элемента

Если находится

Иначе – выводим соответствующее сообщение

выводим разность и соответствующее сообщение



**Задача 6.** Дан массив из 10 случайных чисел из интервала от -10 до 10. Найти номер первого отрицательного элемента (Он обязательно существует)

*// заполнить массив*

**i:=1;**

**while (a[i]>=0) do i:=i+1;**

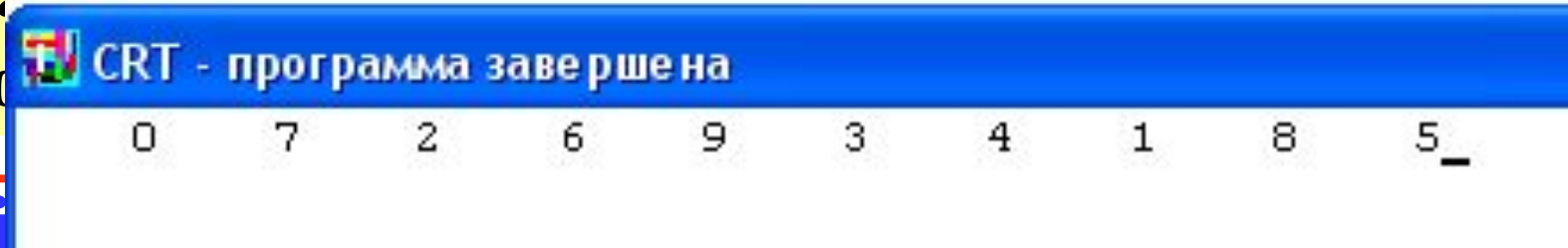
**write('Первое отрицательное ',a[i]:4);**

Пока элемент массива  
положителен или ноль  
берем следующий  
элемент



```
CRT - программа завершена
 3   4   5  -1   3  -9  -4  -4  -2  -6
Первое отрицательное  -1_
```

**Задача 7.** Заполнить массив из 10 элементов случайными числами из интервала [0; 10) так чтобы числа не повторялись.



**For i := 0 to 9 do**

**begin**

**repeat**

**flag:=false;**

**a[i]:=random(10);**

**for j:=1 to i-1 do if a[i]=a[j] then flag:=true;**

**until flag=false;**

**write(a[i]:5)**

**end;**

ЦИКЛ ОТВЕЧАЮЩИЙ

за

Цикл возвращающий процесс назад если такое число уже было

Цикл в котором проверяется не встречалось ли такое число

