Одномерные массивы

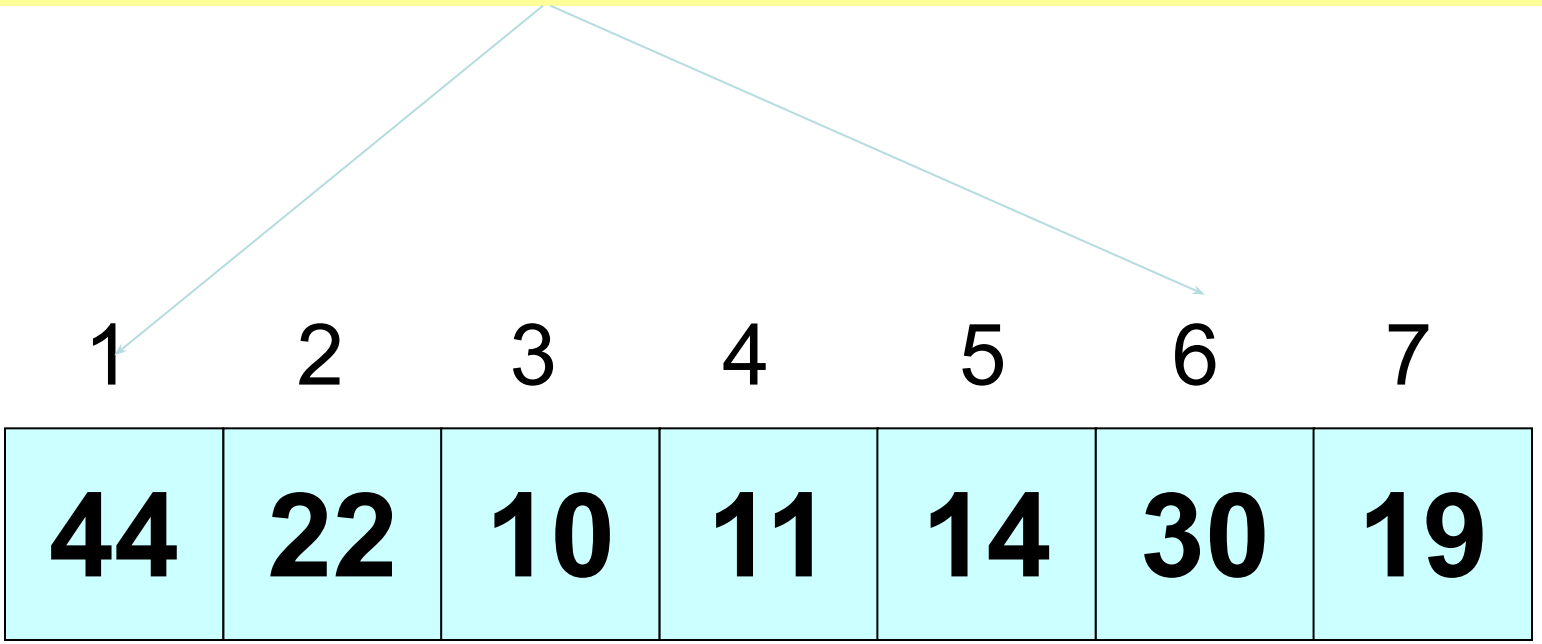
***Массив -** это структура, представляющая собой упорядоченную совокупность элементов одного типа, объединенных одним именем.*

Каждому массиву, используемому в программе, выделяется место в памяти.

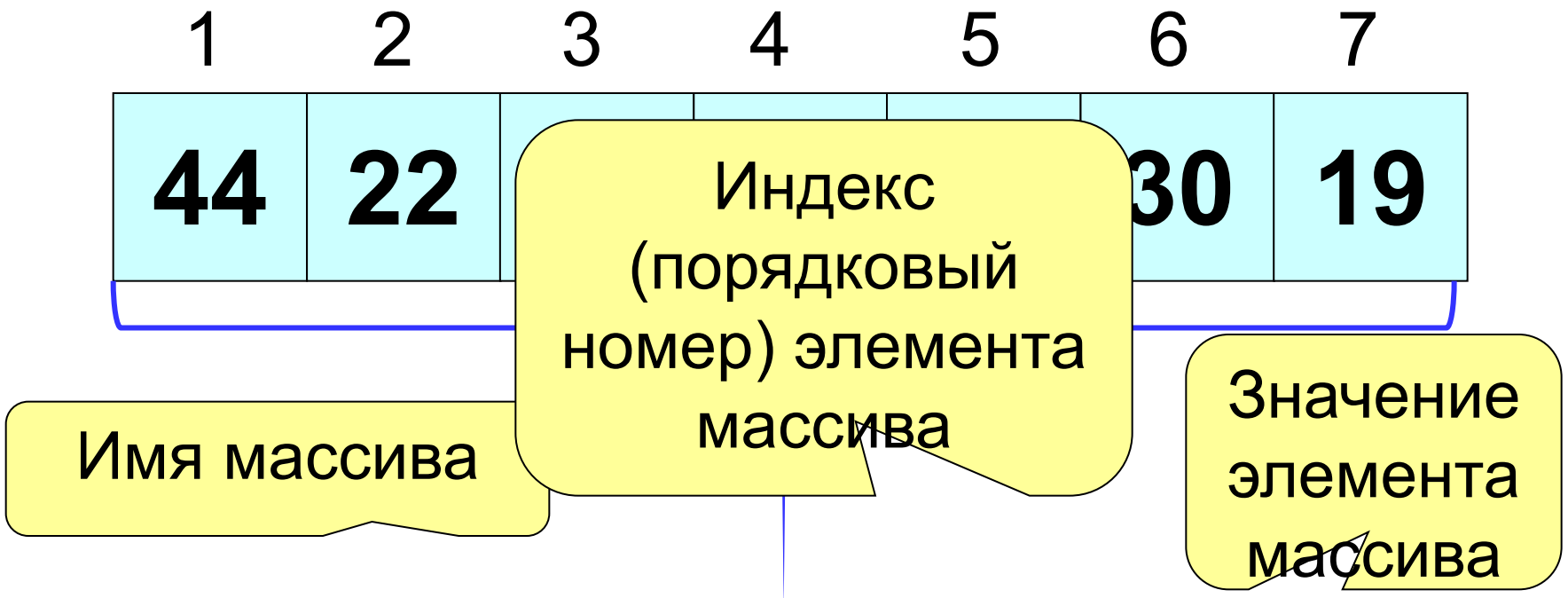
Массиву отводится не одна ячейка, а последовательность расположенных друг за другом ячеек, в каждую из которых записывается значение соответствующего элемента.

44	22	10	11	14	30	19

Для получения доступа к элементу массива используется индекс



Индекс массива - это величина целого типа, характеризующая положение элемента в массиве.


1 2 3 4 5 6 7
44 22 30 19
Индекс (порядковый номер) элемента массива
Имя массива
Значение элемента массива

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

Объявление массивов

Зачем объявлять?

- определить имя массива
- определить тип массива
- определить число элементов
- выделить место в памяти

Массив целых чисел:

```
var A: array[ 1 .. 5 ] of integer ;
```

Размер через константу:

```
const N=5;
var A: array[1..N] of integer;
```

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;

Объявление массивов

Массивы других типов:

```
var X, Y: array [1..10] of real;
    C: array [1..20] of char;
```

Другой диапазон индексов:

```
var Q: array [0..9] of real;
    C: array [-5..13] of char;
```

Индексы других типов:

```
var A: array ['A'..'Z'] of real;
    B: array [False..True] of integer;
...
   A['C'] := 3.14259*A['B'];
   B[False] := B[False] + 1;
```

```
Const
  m=10;
Var  a : array [1.. m] of integer;
```

Количество элементов можно описать в разделе констант

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

Что неправильно?

```
var a: array [1..10] of integer;
...
  A[5] := 4.5;
```

```
var a: array ['a'..'z'] of integer;
...
  A['b'] := 15;
```

```
var a: array [0..9] of integer;
...
  A[10] := 'x';
```

Массивы

Объявление:

```
const N = 5;
var a: array[1..N] of integer;
    i: integer;
```

Ввод с клавиатуры:

```
for i:=1 to N do begin
  write('a[', i, ']=');
  read ( a[i] );
end;
```

```
a[1] = 5
a[2] = 12
a[3] = 34
a[4] = 56
a[5] = 13
```

? Почему write?

Вывод на экран:

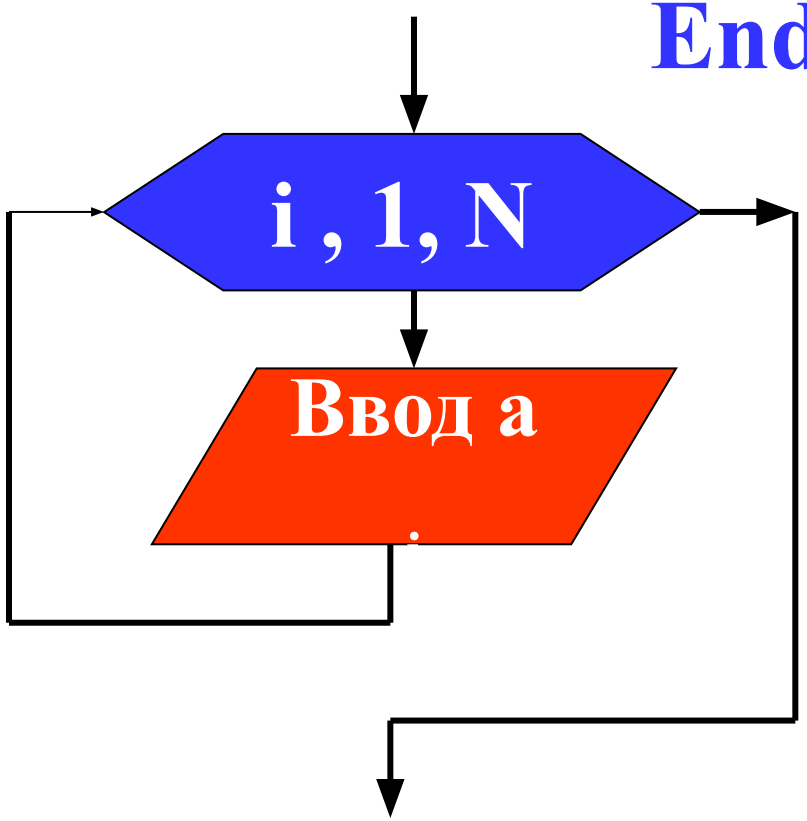
```
for i:=1 to N do a[i]:=a[i]+1;
```

```
writeln('Массив A:');
for i:=1 to N do
  write(a[i]:4);
```

```
Массив A:
   6   13   35   57   14
```

1. Заполнение массива с клавиатуры

```
For i:=1 to N do begin
        Write(' a [ ' , i , ' ] = ' );
        ReadLn ( a [ i ] );
End;
```



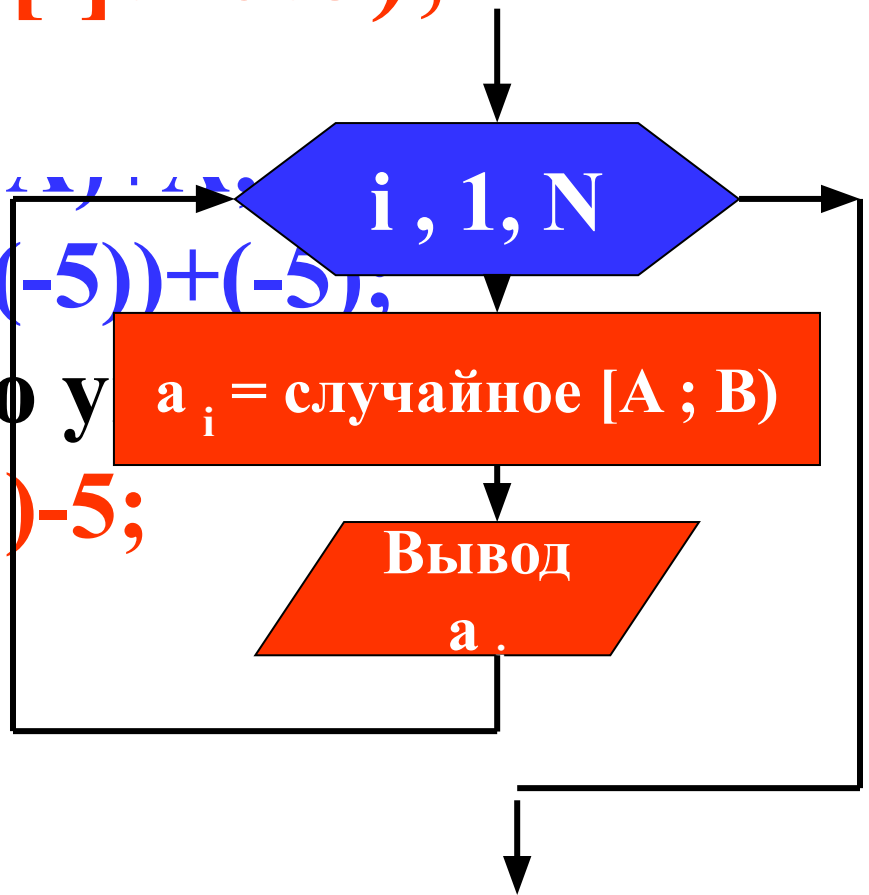
2. Заполнение массива случайными числами.

```
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);
```

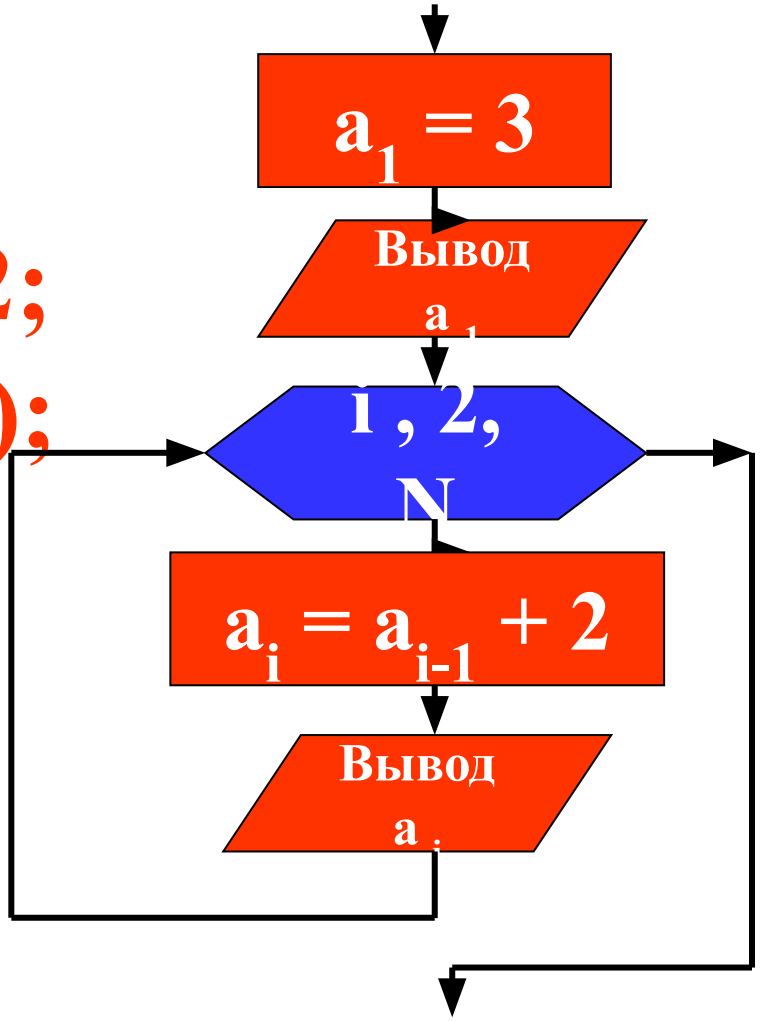
такую запись нужно у

```
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;
```

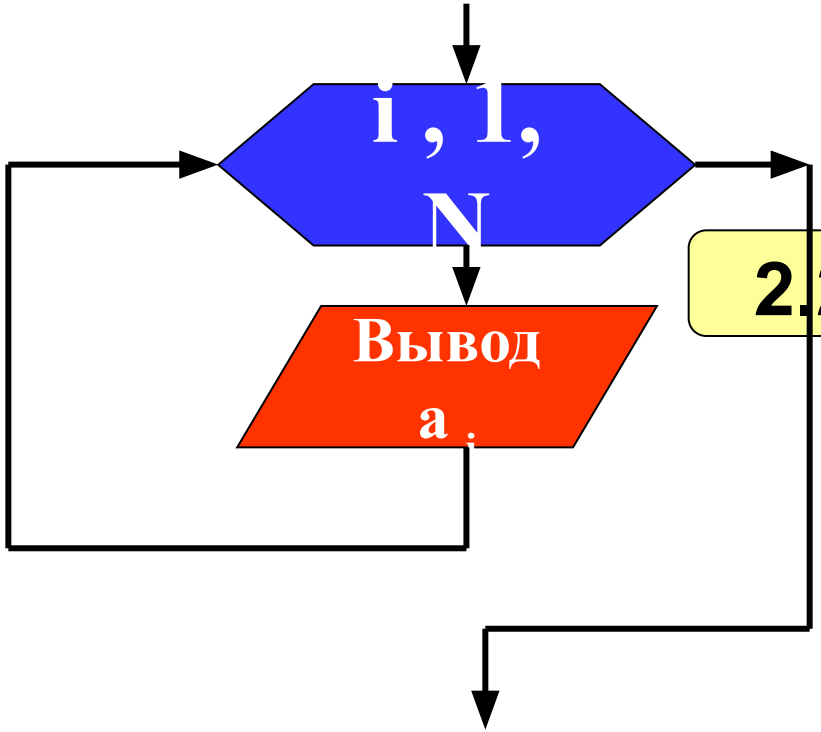


а) в столбик:

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

б) в строку

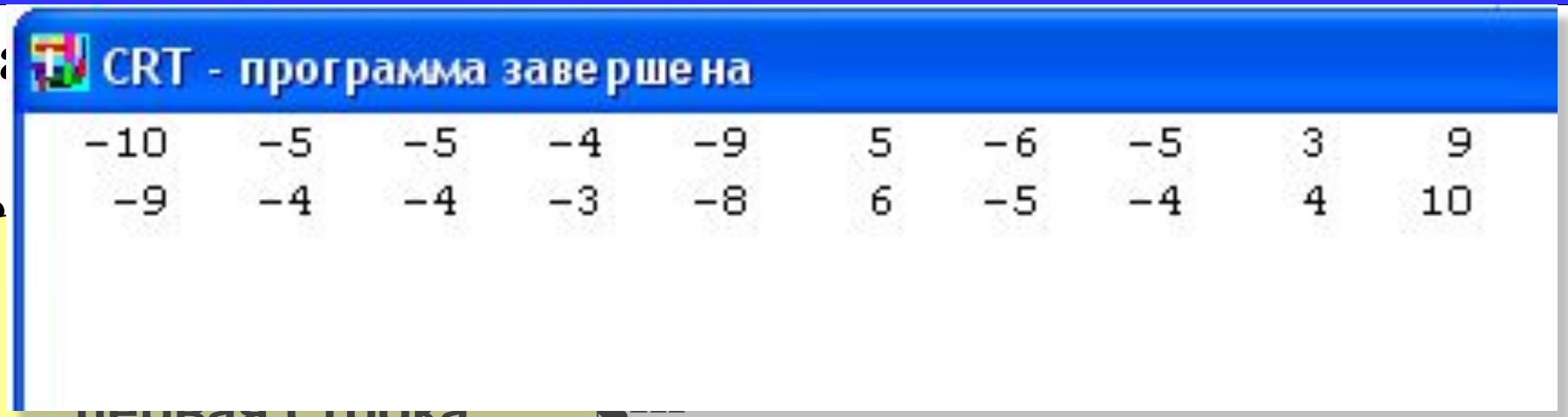
```
For i:=1 to N do Write(a[i]:4:2);
```




```
2.23
3.19
1.44
4.93
5.58
```

2.23 3.19 1.44 4.93 5.58

Вывод массива в две строки



CRT - программа завершена

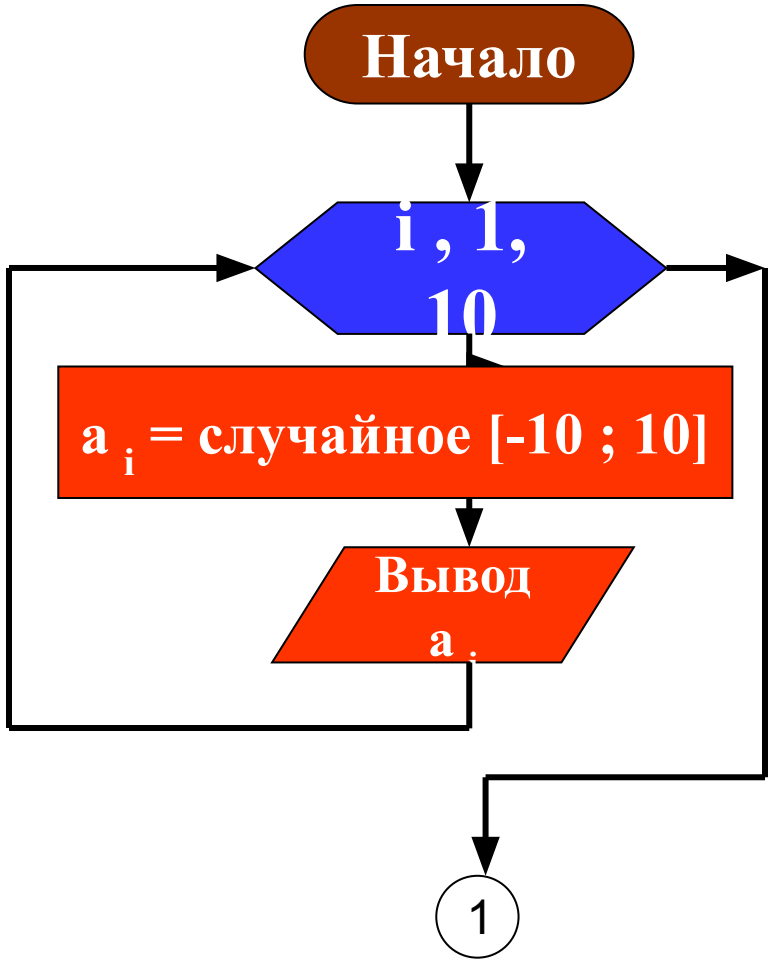
```
-10   -5   -5   -4   -9    5   -6   -5    3    9
 -9   -4   -4   -3   -8    6   -5   -4    4   10
```

первая строка закончена, переход на вторую

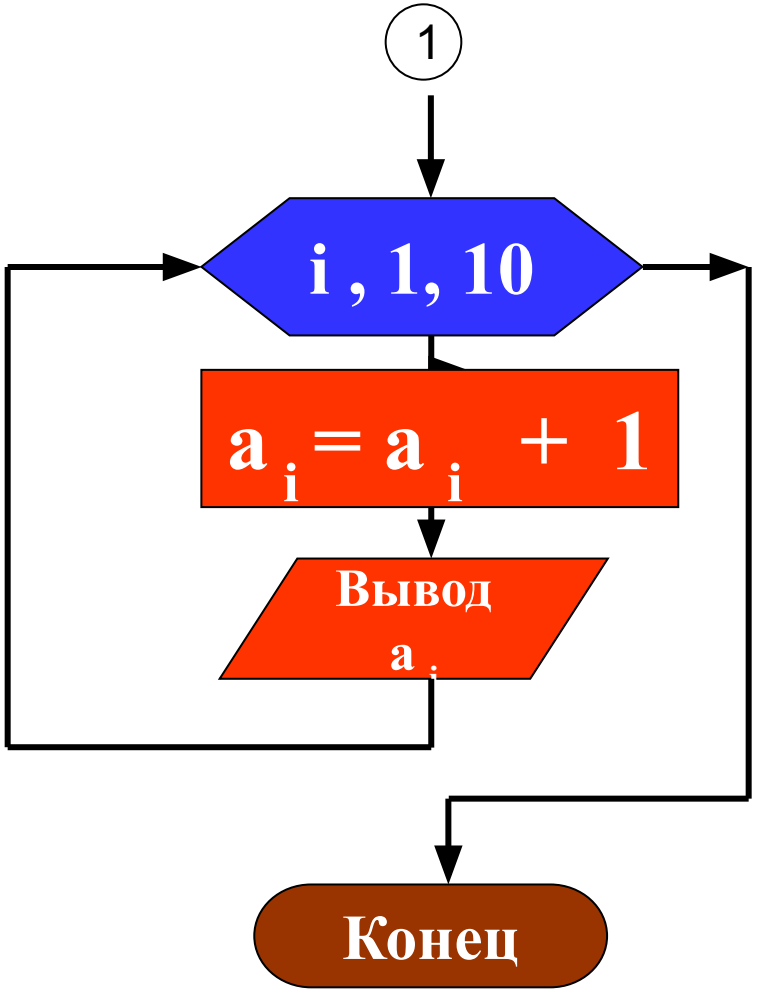
```
                    i]:= random(21)-10;
                    rite(a[i]:5);
                    end;
writeln;
for i:=1 to 10 do begin
                    a[i]:=a[i]+1;
                    write(a[i]:5);
                    end;
end.
```

Оператор вывода без LN, вывод осуществляется в строку

Вывод массива в две строки

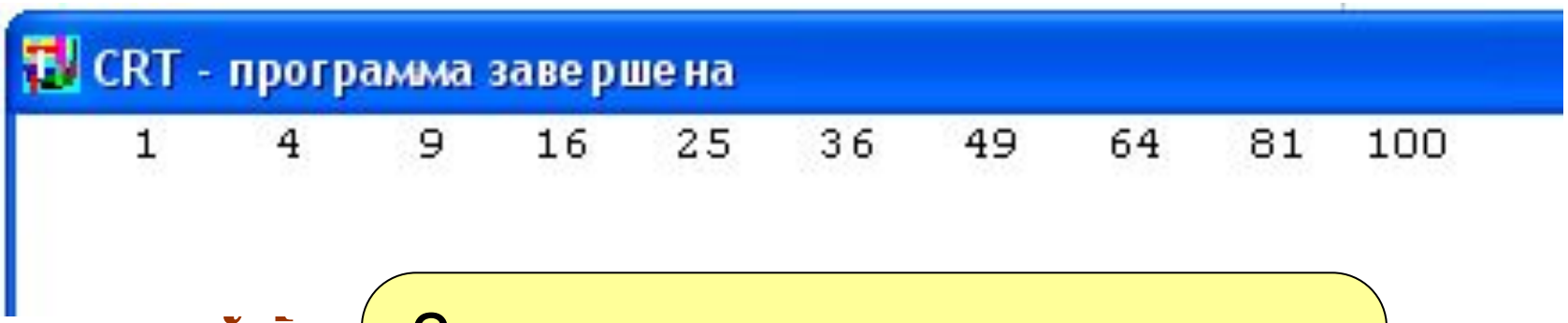


Пустой Writeln между циклами обычно в блок-схеме не изображают.



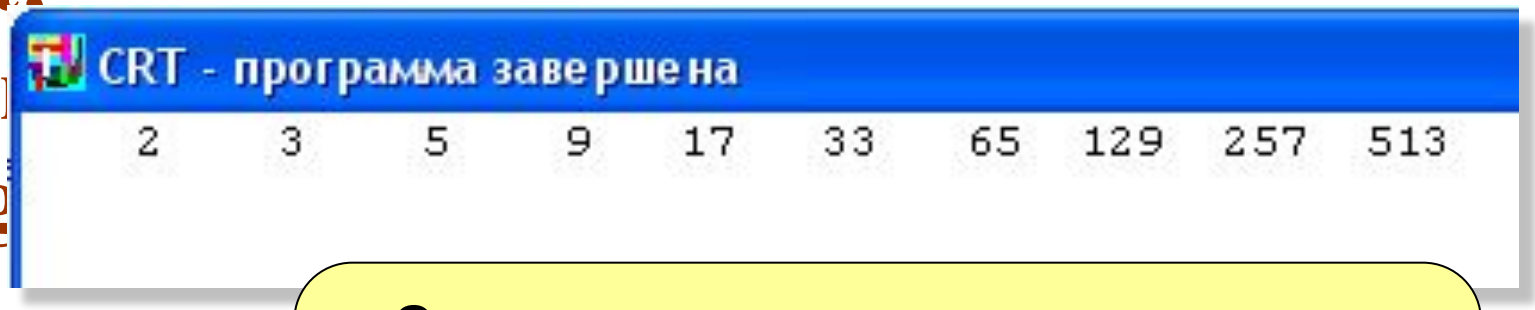
Разрыв блок-схемы, если она целиком на лист не убирается

```
Prog
Uses
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:ar
    i:integ
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

Задания

«3»: Ввести с клавиатуры массив из 5 элементов, умножить их на 2 и вывести на экран.

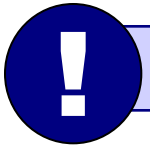
Пример:

```
Введите пять чисел:
4   15    3    10    14
Результат: 8 30 6 20 28
```

«4»: Ввести с клавиатуры массив из 5 элементов, найти среднее арифметическое всех элементов массива.

Пример:

```
Введите пять чисел:
4   15   3  10    14
среднее арифметическое 9.200
```



При изменении N остальная программа не должна изменяться!

Задания

«5»: Ввести с клавиатуры массив из 5 элементов, найти минимальный из них.

Пример:

```
Введите пять чисел:
4   15    3    10    14
минимальный элемент 3
```

Практикум: заполнение массива

«3»: 1. Заполните массив A нулями.

2. Заполните массив A первыми N натуральными числами, начиная с 1.

3. Заполните массив A первыми N натуральными числами, начиная с X (ввести X с клавиатуры).

«4»: 4. Заполните массив A первыми N натуральными числами, начиная с X (ввести X с клавиатуры).

5. Заполнить массив A первыми N числами Фибоначчи. Первые два числа Фибоначчи равны единице, а каждое последующее число Фибоначчи вычисляется как сумма двух предыдущих.

«5»: 6. Заполните массив степенями числа 2, так чтобы последний элемент массива был равен 1, а каждый предыдущий был в 2 раза больше следующего. Например: 32 16 8 4 2 1

7. Заполните массив целыми числами, так чтобы средний элемент массива был равен X, слева от него элементы стоят по возрастанию, а справа – по убыванию (ввести X с клавиатуры). Соседние элементы отличаются на единицу. Например: 1 2 3 2 1.

Практикум: изменение элементов массива

«3»:

1. Увеличить все элементы массива A на 1.
2. Умножить все элементы массива A на 2.
3. Возвести в квадрат все элементы массива A.

«4»:

4. Увеличить на 4 все элементы в первой половине массива A (считать, что в массиве чётное число элементов).
5. Разделить на 2 все элементы массива A, кроме первого и последнего (считать, что в массиве есть, по крайней мере, два элемента и все элементы чётные).

«5»:

6. Умножить на 3 все элементы во второй половине массива A (считать, что в массиве чётное число элементов).
7. Найти среднее арифметическое всех элементов массива A.

© К. Поляков, 2006-2011

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

Нахождение суммы элементов массива

Предполагаем, что сумма равна нулю

```
s:=0;
For i:=1 to N do If (условие) then s:=s+ a[i];
WriteLn ('Сумма = ', s:10:5);
```

Нахожде

Если элемент подходит по условию, добавляем произведению

Предполагаем, что произведение равно единице

, добавляем его к сумме

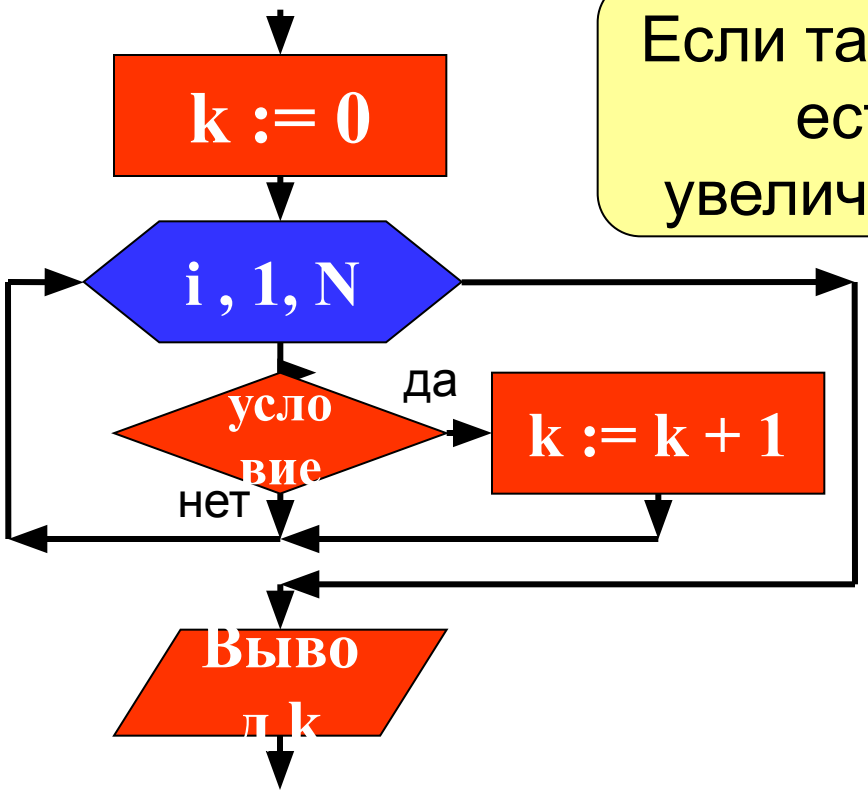
```
p:=1;
For i:=1 to N do If (условие) then p:=p* a[i];
WriteLn('Произведение = ', p:10:5);
```

По элементов у условию:

Предполагаем, что таких элементов нет

```
k:=0;
For i:=1 to N do If (условие) then k:=k+1;
WriteLn('Кол-во=',k:10);
```

Если такой элемент есть, то К 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;
```

Предпола
сумма ра

Организуе
15 повт

Определяем число случайным образом

Добавляем к сумме

Выводим его на экран

Нахождение суммы элементов массива

Находим сред
арифметичес

Выводим его на экран

Предполагаем, что таких эл

Организуем цикл из 15 повторений

```
Writeln;
s:=s/15;
Writeln(
e = ',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 К на единицу

Выводим К на экран

Нахождение максимального элемента в массиве и его номера:

```
max:=a[1];
k:=1;
for i:=2 to N do if (a[i]>max) then begin
  max:=a[i];
  k:=i;
end;
WriteLn('Max=',max:10:5);
WriteLn('Номер=',k:5);
```

Предпо… наиболь… стоит на …

Если находится больший элемент …е

Когда массив закончился, выводим наибольший элемент и его номер

Он становится максимальным

Иногда в задаче достаточно определить индекс

Предполагаем, что
элемент стоит
ом месте

ma

fo

wr

wr

CRT - программа завершена

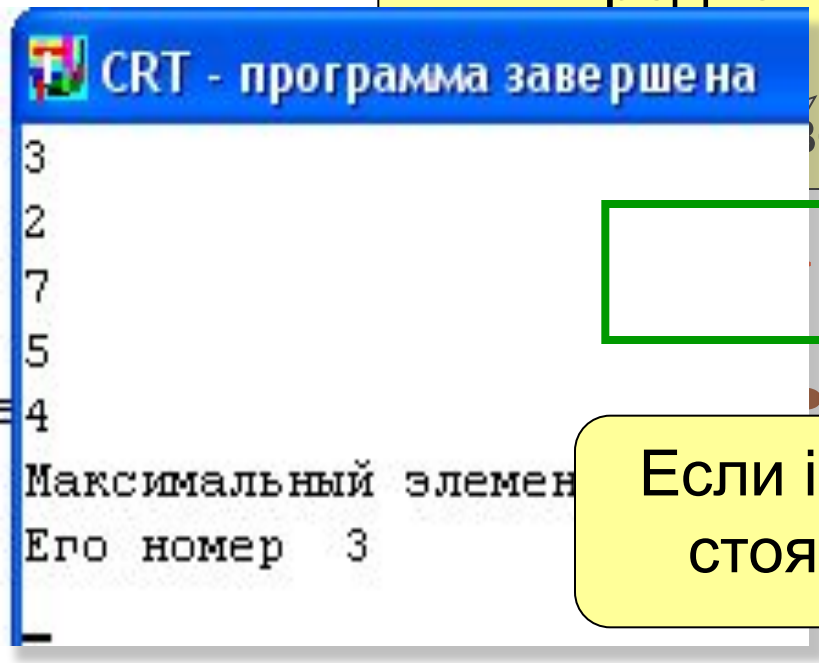
```
3
2
7
5
4
Максимальный элемен
Его номер  3
```

a[max] then max:=i;

ый элемент ',a[max]);

Если i-ь
стоящ

то, запоминаем его индекс в переменной max



Решение задач

1.Дан массив из 100 элементов заполненный случайными целыми числами из интервала [-100, 100]. Найдите наибольший элемент среди четных элементов массива.

2. Дан массив A размера N, введенный с клавиатуры и заполненный нулями и единицами. Найти наибольшее количество идущих подряд единиц.

3. Даны два массива A и B размера N, введенные с клавиатуры. Сформировать новый массив C по правилу: на нечетных местах стоят элементы массива A, а на четных местах стоят элементы массива B.

4. Дан массив A размера N, заданный случайным образом [-100, 100]. Сформировать новый массив B из отрицательных элементов массива A.

Вставка элементов удовлетворяющих условию в другой массив. Пусть нам дан массив с им… элементы нужно в…

Порядковый номер элемента в ма…

Если эл… удовл… ус…

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]…) then begin
        j:=j+1;
        b[j]:=a[i];
K:=j;
```

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

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 с индексом P

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

100

	9	12	11	14	7	

p N

Написать программу к задаче:

Сформировать массив из 20 элементов случайным образом на отрезке [-100,100]. Вывести его на экран. Вставить перед каждым отрицательным элементом массива 0. Вывести измененный массив на экран.

Задача 7. Вставить перед каждым отрицательным элементом массива 0.

```
i:=1; n:=10;
while i<=n do
begin
      if a[i]<0 then begin
          for j:=n downto i do  a[j+1]:=a[j];
           a[i]:=0; i:=i+2; n:=n+1;
       end
     else i:=i+1;
end;
```

Удаление элемента с номером p из массива:

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

Пе… …аем эл…
ме…
пр…

Уменьшаем на 1 …
количество элемент…

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

	100	9	12	11	14	7
	p					N

Написать программу к задаче:

Сформировать массив из 20 элементов случайным образом на отрезке [-100,100]. Вывести его на экран. Удалить из массива все нулевые элементы .

Удалить из массива все нулевые элементы: 1 способ

```
j:=1;
While j<=N do
begin
  If a[j]=0 then begin
                   k:=j
                   for i:=k to (N-1) do a[i]:=a[i+1];
                   N:=N-1;
                     end
              else  j:=j+1;
   end;
```

Удалить из массива все нулевые элементы: 2 способ

```
For i:=N downto 1 do
  If a[i]=0 then begin
                  for k:=i to (N-1) do a[k]:=a[k+1];
                    N:=N-1;
                     end;
```

Поиск трех минимальных элементов в неповторяющемся массиве

```
min1 :=maxint; min2 :=maxint; min3 := maxint;
for i:=1 to N do begin
  if p[i] < min1 then begin { новый min1 }
    min3 := min2; min2 := min1;
    min1 := p[i];   end
 else if p[i] < min2 then begin { новый min2 }
      min3 := min2;
      min2 := p[i];     end
    else if p[i] < min3 then { новый min3 }
        min3 := p[i];
end;
```

```
min1:=maxint; min2:=maxint; min3:= maxint;
for i:=1 to N do begin
  if a[i] < min1 then begin
    min3:=min2; min2:=min1; min1:=a[i];   end
 else
 if (a[i] < min2) and (a[i]<>min1) then begin
                min3:= min2; min2:=a[i]; end
else if (a[i] < min3) and (a[i]<>min2) and (a[i]<>min1)
then
          min3 := a[i];
end; writeln;
writeln('min1=',min1,' min2=',min2,' min3=',min3);
```

```
// 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('Ар                                    ');
        End
else
Writeln('Не арифмети                                  );
```

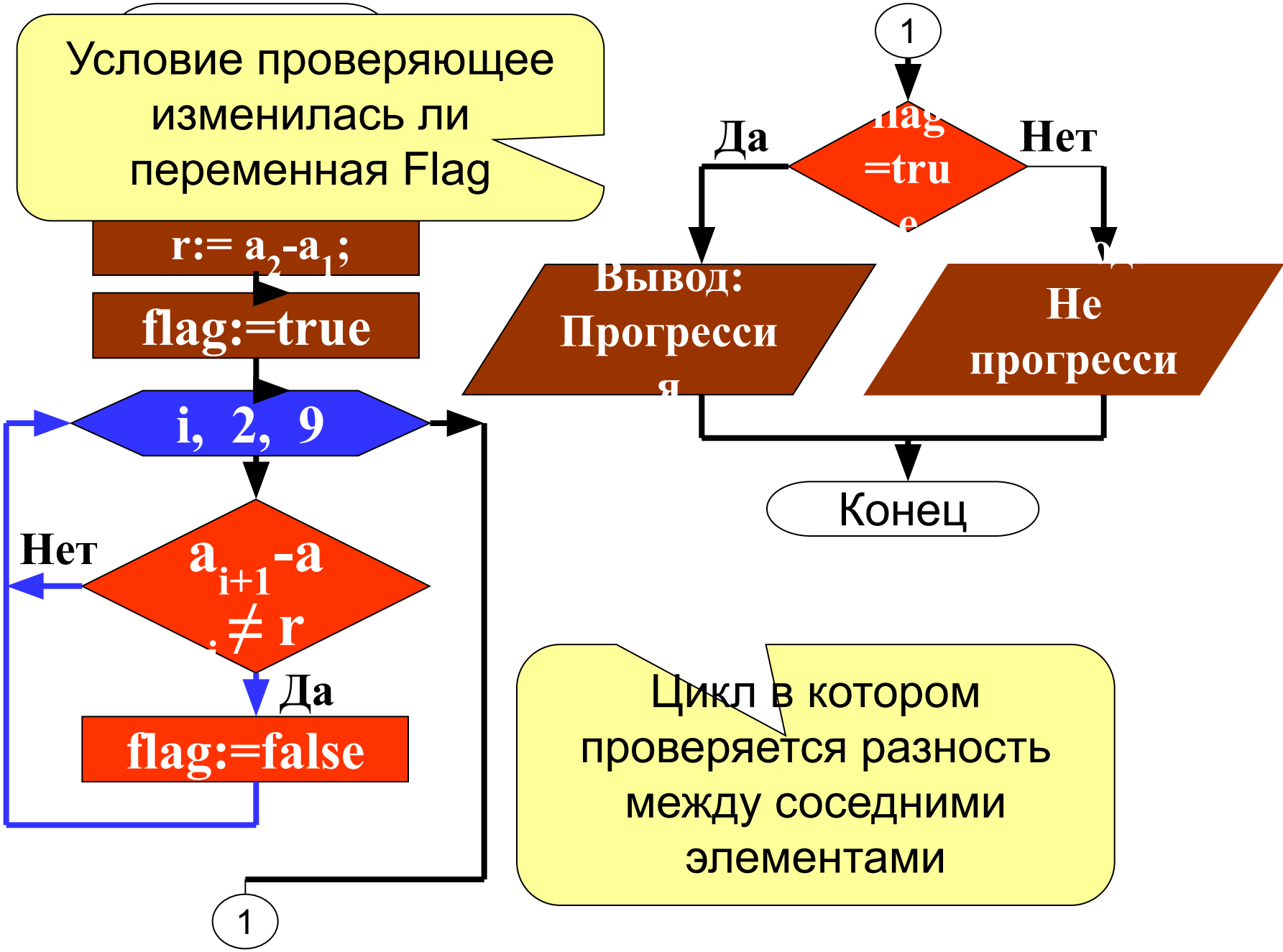
Находим разность

Организуем цикл от второго до предпоследнего элемента

Если значение истинно

Иначе – выводим соответствующее сообщение

Если находится ...итов, у ...угая ...ото не ...то выводим разность и соответствующее сообщение


Условие проверяющее изменилась ли переменная Flag
r:= a2-a1;
flag:=true
i, 2, 9
Нет
ai+1-a
≠ r
Да
flag:=false
1
1
Да
=tru
e
Нет
Вывод:
Прогресси
я
Не
прогресси
Конец
Цикл в котором проверяется разность между соседними элементами

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

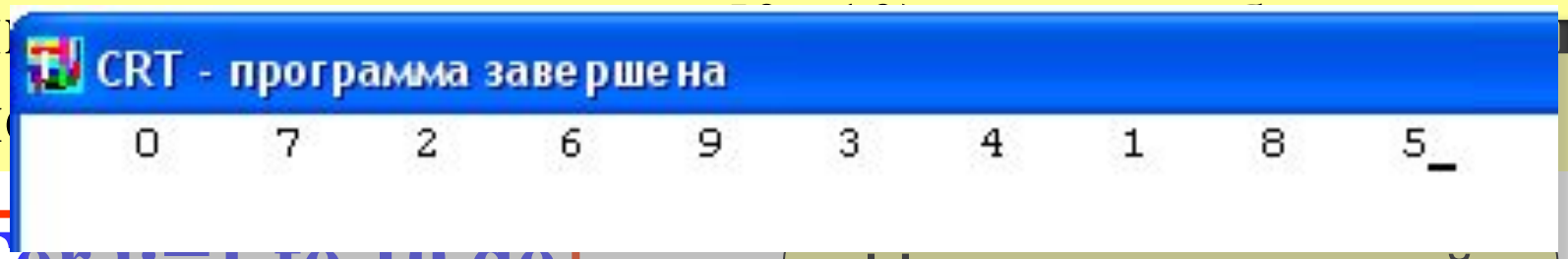
```
// заполнить массив
i:=1;
while (a[i]>=0) do i:=i+1;
write('Первое отрицательное ',a[i]:4);
```

Пока элемент массива положителен или ноль берем следующий элемент

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

```
For i:=1 to 10 do
    begin
      repeat
        r:=false;
        a[i]:=random(10);
        for j:=1 to i-1 do if a[i]=a[j] then r:=true;
      until r=false;
      write(a[i]:5);
    end;
```

***Задача 7*. Заполнить массив из 10 элементов случайными ч… не п…**

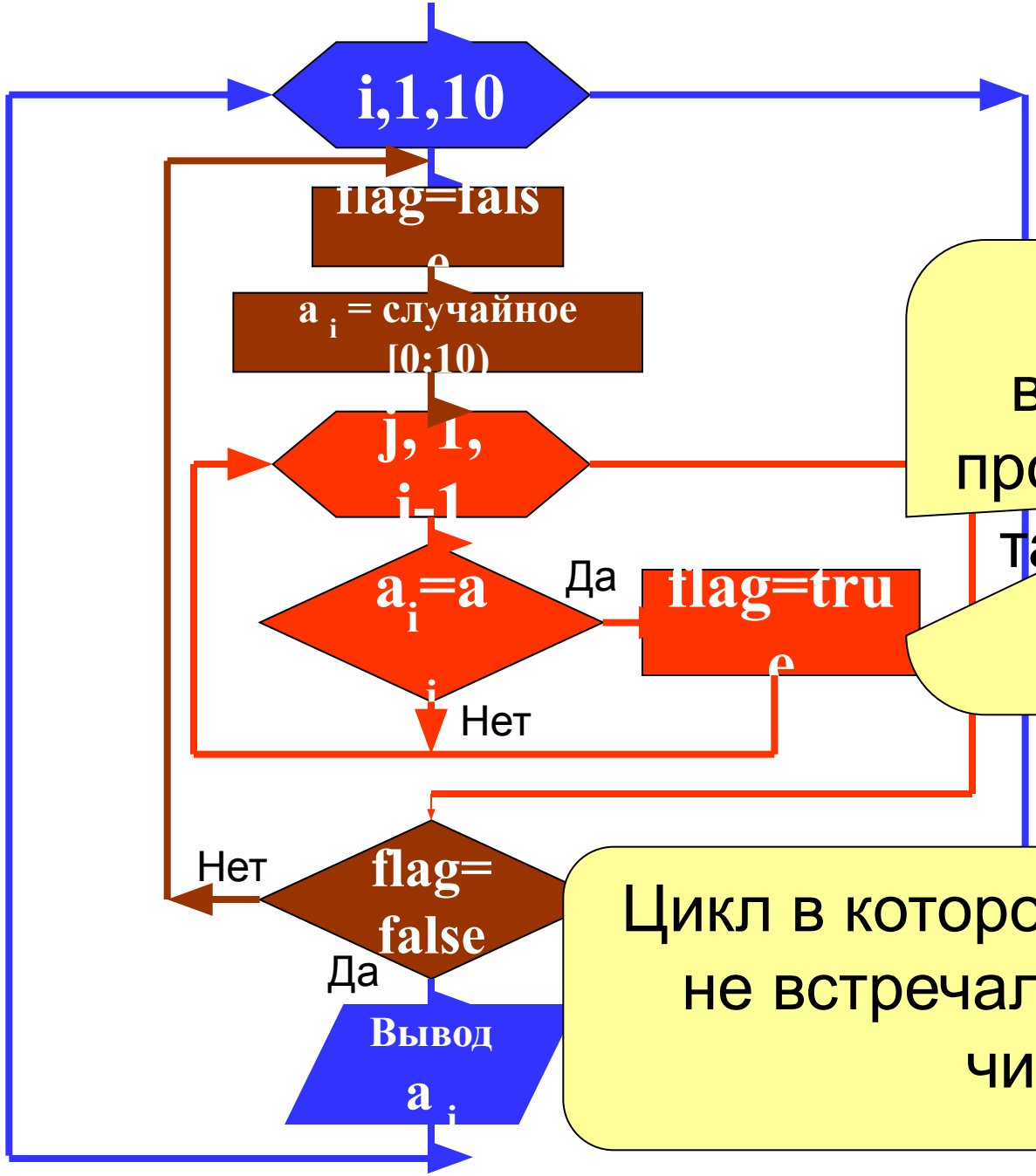



```
For i:=1 to 10 do
  begin
    repeat
      flag:=false;
      a[i]:=random(10);
      for j:=1 to i-1 do if a[i]=a[j] then flag:=true;
    until flag=false;
    write(a[i]:5)
  end;
```

Цикл отвечающий за…

Цикл возвращающий процесс назад если такое число уже было

Цикл в котором проверяется не встречалось ли такое число



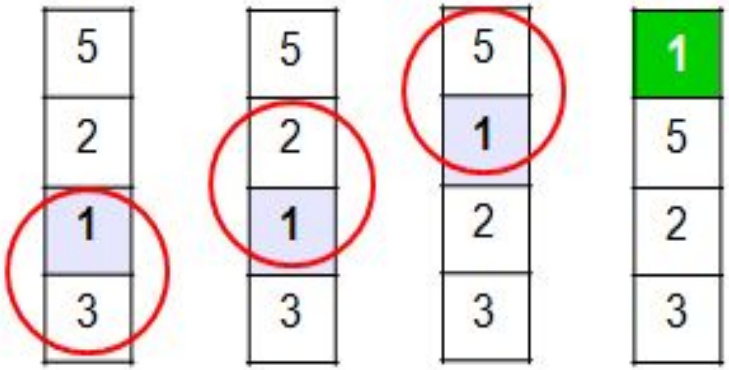
Цикл отвечающий за количество чисел в массиве

воз
про
такое число уже было

Цикл в котором проверяется не встречалось ли такое число

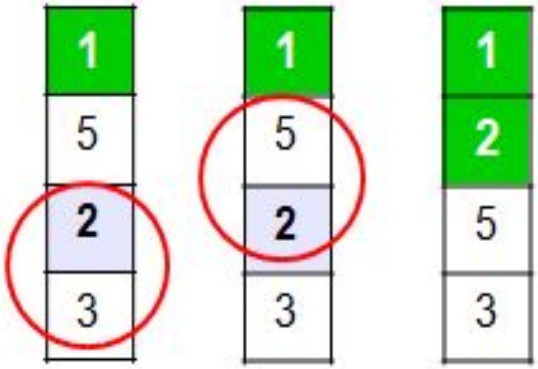
Идея – самый маленький («легкий» элемент перемещается вверх («всплывает»).

1-ый проход

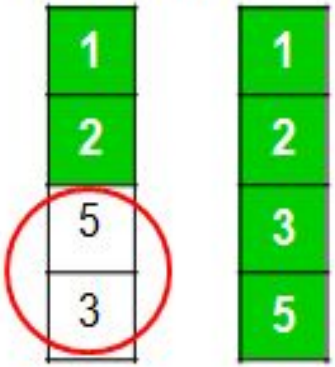


- начиная снизу, сравниваем два соседних элемента; если они стоят «неправильно», меняем их местами
- за 1 проход по массиву **один** элемент (самый маленький) становится на свое место

2-ой проход



3-ий проход



Для сортировки массива из N элементов нужен N-1 проход (достаточно поставить на свои места N-1 элементов).

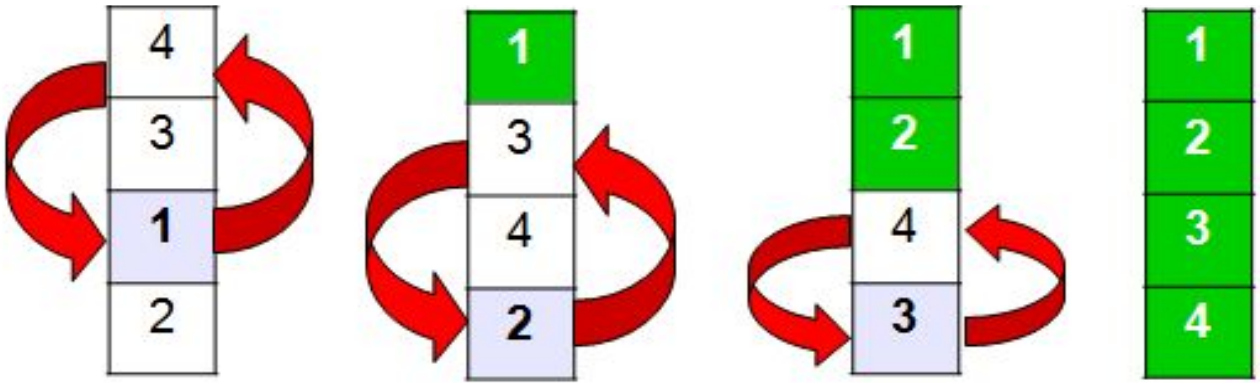
Сортировка по убыванию

```
for j:=1 to n-1 do begin
    for i:=1 to n-j do
   if a[i]<a[i+1] then begin c:=a[i];
                             a[i]:=a[i+1];
                             a[i+1]:=c;
                       end;
```

Метод выбора (минимального элемента)

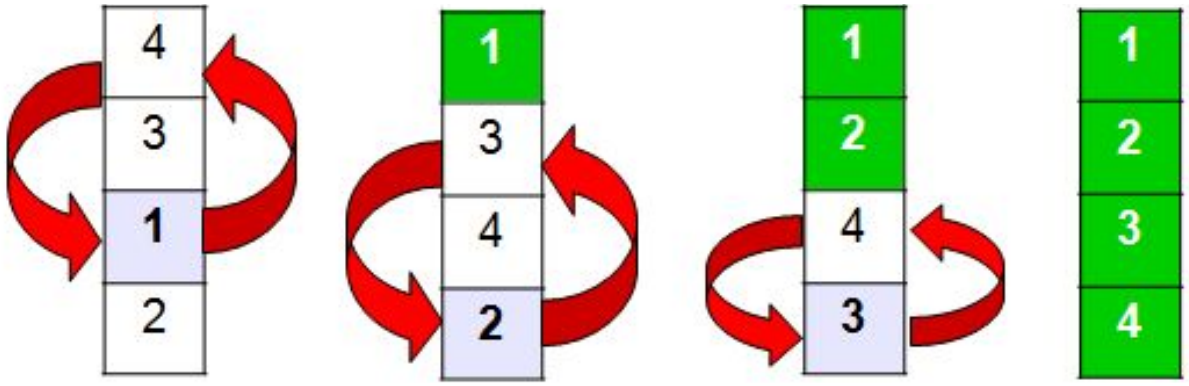
Идея: найти минимальный элемент и поставить его на первое место.

```
for i:=1 to N-1 do begin
  { найти номер nMin минимального элемента
    из A[i]..A[N] }
  if i <> nMin then begin
    { поменять местами A[i] и A[nMin] }
  end
end;
```



Метод выбора (минимального элемента)

```
for i:=1 to N-1 do begin
  nMin:=i;
  for j:=i+1 to N do
    if A[j]<A[nMin] then
      nMin:=j;
  if i <> nMin then begin
    { поменять местами A[i] и A[nMin] }
  end
end;
```



Сортировка вставками

Идея: сначала упорядочиваются два элемента, затем делается вставка третьего элемента относительно двух первых. Затем вставка четвертого относительно трех первых и т.д.

```
for i:=2 to N do
          begin
            k:=a[i];  j:=i-1;
            while (j>=1) and (a[j]>k) do
              begin
                a[j+1]:=a[j];  j:=j-1;
              end;
            a[j+1]:=k;
          end;
```

Задачи

«А»: Массив содержит четное количество элементов. Напишите программу, которая сортирует первую половину массива по возрастанию, а вторую – по убыванию. Каждый элемент должен остаться в «своей» половине.

Пример:

```
Массив:
5 3 4 2 1 6 3 2
После сортировки:
2 3 4 5 6 3 2 1
```

Задачи

«B»: Напишите программу, которая сортирует массив и находит количество различных чисел в нем.

Пример:

```
Массив:
5 3 4 2 1 6 3 2 4
После сортировки:
1 2 2 3 3 4 4 5 6
Различных чисел: 5
```

«C»: Напишите программу, которая сравнивает число перестановок элементов при использовании сортировки «пузырьком» и методом выбора. Проверьте ее на разных массивах, содержащих 1000 случайных элементов, вычислите среднее число перестановок для каждого метода.

Общая идея алгоритма:

1. Берем i-ый элемент
2. Последовательно сравниваем его со всеми элементами справа
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…

Сортировка выбором

элементов масси

Перебираем элементы от первого до предпоследнего

```
For i:=1 to N-1 do
do
a[i]>a[j]) then begin
p:=a[i];
a[i]:=a[i];
```

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

ставш
находи
еньш
й …

Сортировка массива:
4; -2; 6; -5
по возрастанию с изменениями происходящими в массиве

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

http://privples.iv-edu.ru/informatica/1/Mater/SBORNIK_ZADAS.doc