Лекция №8
Обработка одномерных массивов.

Определение одномерного массива.

Массив - это структурированный тип данных, представляющий собой последовательность однотипных элементов, имеющих общее имя и снабженных индексами (порядковыми номерами).

Массивы удобно использовать для хранения однородной по своей природе информации, например, таблиц и списков.

Например, в виде одномерного массива строк можно задать список студентов:

```
( 'Иванов', 'Петров', 'Сидоров', 'Песков', 'Петренко'),
```

А в виде массива вещественных чисел – рост этих студентов (в см):

(178,2 172,3 200,5 185,2 169)

Последовательность

```
('Иванов'    178,2    'Петров'    'Сидоров'
25    'Песков', 'Петренко')
```

не может являться массивом.

Элементами массива могут быть данные любого типа.

Индекс - это выражение целого типа (*integer, byte)*, определяющее положение элемента в массиве.

Размерность массива - это количество элементов в массиве.

Элемент массива в языке Delphi обозначается следующим образом:

<имя массива>[*<индекс>*].

Здесь *<имя массива>* - правильный идентификатор.

Например, *x*[3] - третий элемент массива *x*.

Описание одномерного массива.

Перед использованием массив должен быть описан. Существует два способа описания массива :

а)

var

< имя массива >: *array* [*< нач. значение индекса >..< кон. знач. инд.>*] *of* *< тип элементов>*;

Например,

```
var
    x:array[1..20] of integer;
```

В памяти компьютера будет отведено место для 20 целых чисел (80 байт). Т.е. максимальная размерность массива в этом случае – 20 элементов.

Описание одномерного массива.

б) *type*

< имя типа > = array [*< нач. значение индекса >..< кон.знач. инд. >*]

of < тип элементов>;

var

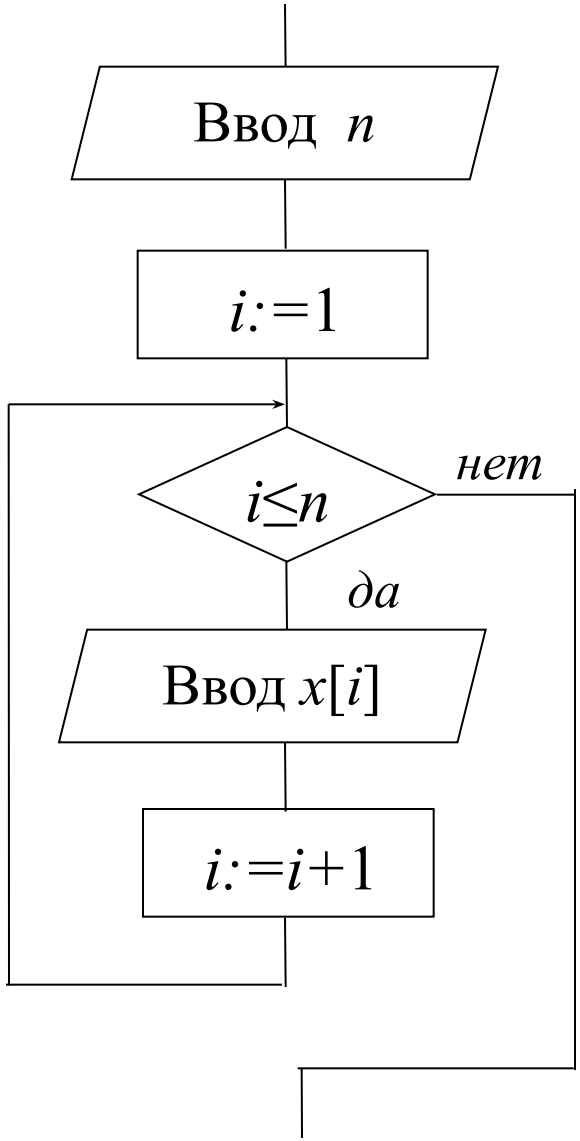
< имя массива >: *< имя типа >*;

Например,

```
type
        massiv = array[1..20] of integer;
    var
        x : massiv;
```

В памяти компьютера будет отведено место для 20 целых чисел (40 байт).

Ввод одномерного массива с использованием ВК StringGrid.



Пусть n - размерность массива;

x - исходный массив;

i - номер текущего элемента массива.

Для реализации этого алгоритма на форме нужно разместить ВК **Edit** для ввода размерности массива и ВК **StringGrid** для ввода значений элементов масива.

Для **StringGrid** нужно установить значение true для опции **goEditing** свойства **Options,**

при вводе в строку для свойства **ColCount** установить значение, равное максимальной размерности массива, а для свойства **RowCount** установить значение 1;

при вводе в столбец установить значение, равное максимальной размерности массива, для свойства **RowCount,** а для свойства **ColCount** установить значение 1**.**

Если x - массив целых чисел (например, x :array[1..20] of integer;), то в программе для ввода используем следующий фрагменты:

а) ввод в строку

```
n:=strtoint(edit1.Text);
for i:=1 to n do
   x[i]:=strtoint(stringgrid1.Cells[i-1,0]);
```

б) ввод в столбец

```
n:=strtoint(edit1.Text);
for i:=1 to n do
   x[i]:=strtoint(stringgrid1.Cells[0,i-1]);
```

Если x - массив вещественных чисел (например, x :array[1..20] of real;), то в программе для ввода используем следующий фрагменты:

а) ввод в строку

```
 n:=strtoint(edit1.Text);
   for i:=1 to n do
     x[i]:=StrToFloat(stringgrid1.Cells[i-1,0]);
```

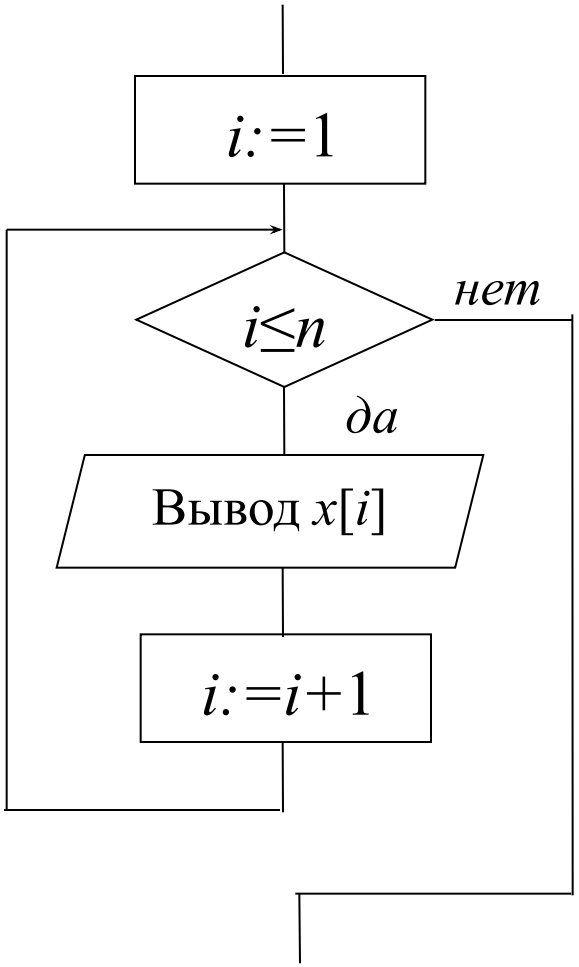
б) ввод в столбец

```
n:=strtoint(edit1.Text);
for i:=1 to n do
   x[i]:=StrToFloat(stringgrid1.Cells[0,i-1]);
```

Для того, чтобы при изменении размерности массива автоматически менялось количество строк или столбцов компонента **StringGrid**, нужно создать процедуру – обработчик события изменения текста в компоненте Edit двойным щелчком на этом компоненте. И набрать код:

```
procedure TForm1.Edit1Change(Sender: TObject);
begin
 if  edit1.text<>'' then
  stringgrid1.ColCount:=StrToInt(edit1.text)//для ввода
                                            //в строку
end;
```

Вывод одномерного массива с использованием ВК StringGrid.



а) Вывод массива целых чисел в столбец

```
for i:=1 to n do
 stringgrid1.Cells[0,i-1]:=
                        IntToStr(x[i]);
```

б) Вывод массива целых чисел в строку:

```
 for i:=1 to n do
   stringgrid1.Cells[i-1,0]:=
                        IntToStr(x[i]);
```

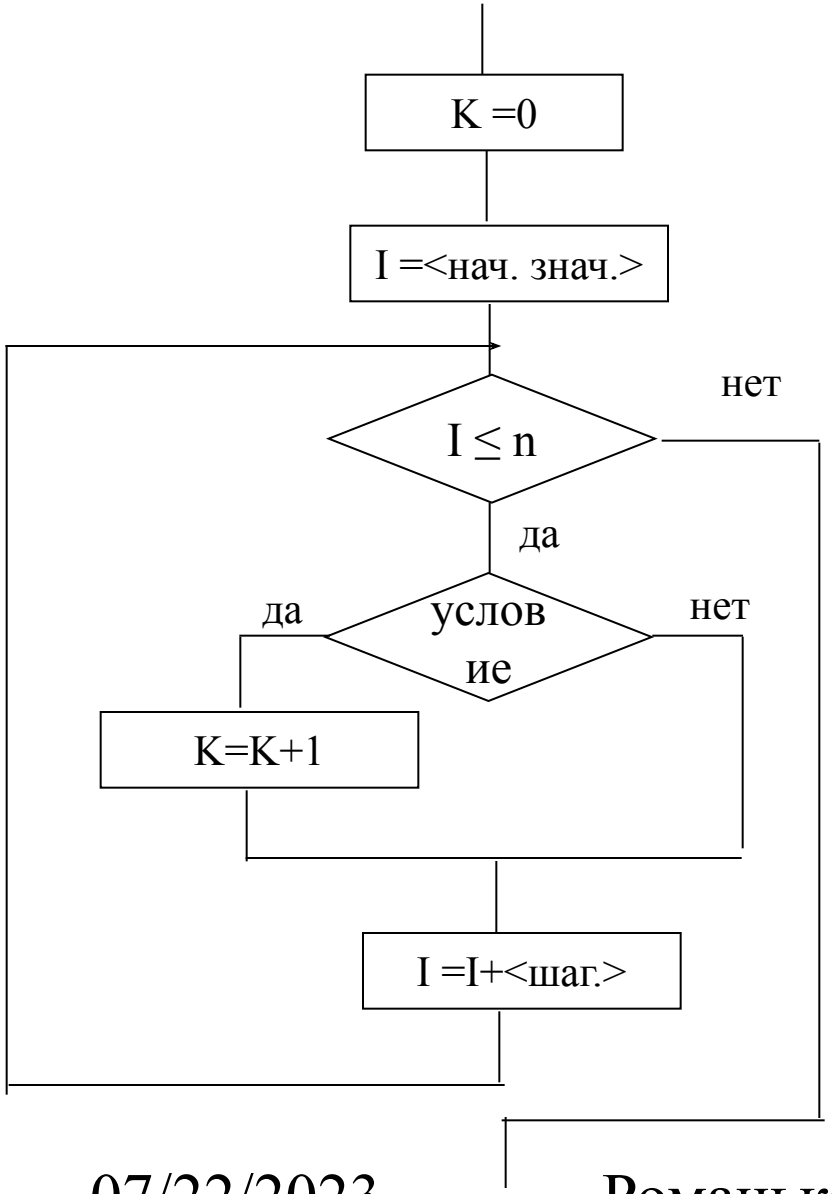
Аналогично осуществляется вывод массива вещественных чисел.

Типовые алгоритмы обработки одномерных массивов.

Вычисление суммы, произведения, количества элементов массива.

Нахождение максимального и минимального элементов массива.

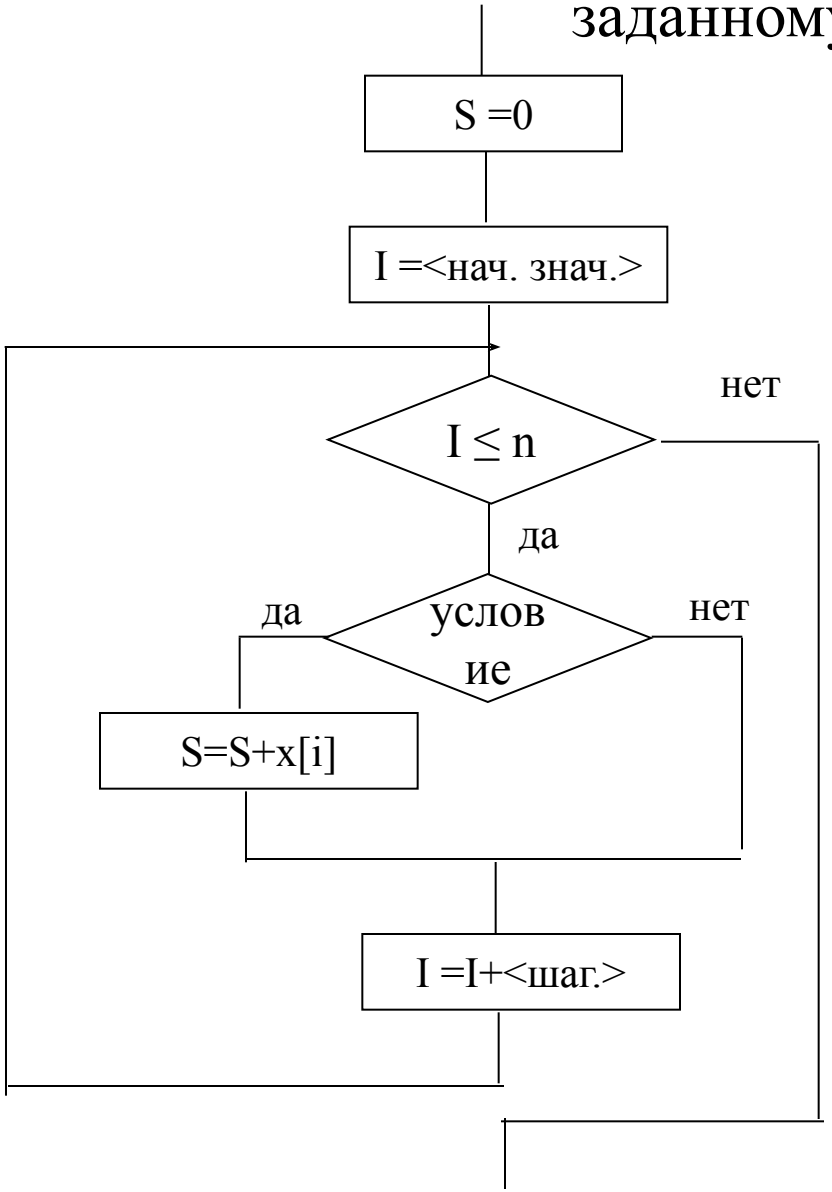
Вычисление количества элементов массива, удовлетворяющих заданному условию.



Реализация в программе:

```
k:=0;
  i:=<нач. Знач.>;
  while i<=n do
   begin
     if <условие> then
       k:=k+1;
      i:=i+<шаг>
    end;
```

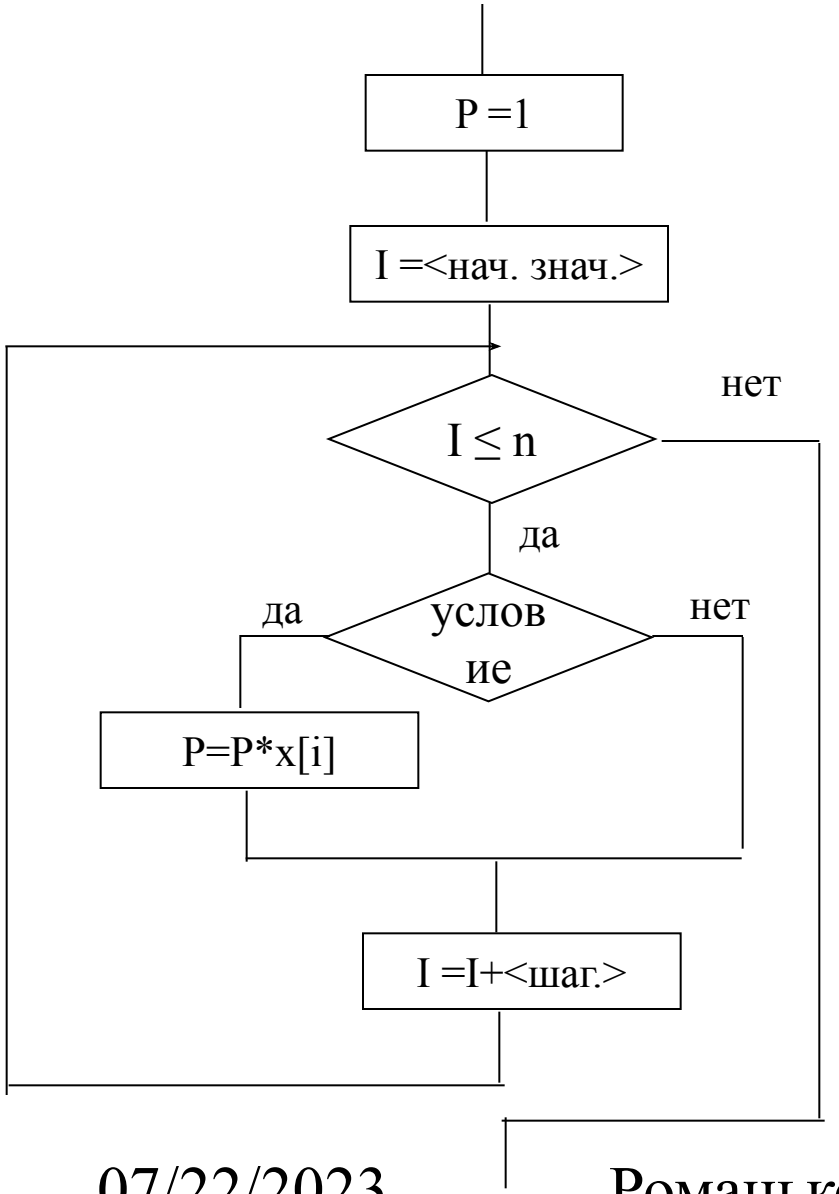
Вычисление суммы элементов массива, удовлетворяющих заданному условию.



Реализация в программе:

```
S:=0;
  i:=<нач. Знач.>;
  while i<=n do
   begin
      if <условие> then
        S:=S+x[i];
       i:=i+<шаг>
    end;
```

Вычисление произведения элементов массива, удовлетворяющих заданному условию




```
P:=1;
i:=<нач. Знач.>;
while i<=n do
 begin
   if <условие> then
     P:=P*x[i];
    i:=i+<шаг>
  end;
```

Пример вычисления суммы, произведения, количества элементов массива

В заданном массиве найти среднее арифметическое отрицательных элементов, стоящих на местах, кратных 3; вычислить произведение элементов, не принадлежащих интервалу (A , B].

Например, пусть задан массив

$$x = (-2 \quad 3 \quad -3 \quad 1 \quad 4 \quad -5 \quad -1 \quad 0 \quad 2 \quad 8 \quad -3 \quad -7)$$

Заданный интервал (-3, 3].

Тогда среднее арифметическое отрицательных элементов с номерами, кратными 3, должно получиться равным

$$(-3 + -5 + -7) / 3 = -5$$

Произведение элементов, не принадлежащих интервалу (-3 , 3]:

$$-3 * 4*(-5)*8*(-3)*(-7) = 10080$$

Введем следующие обозначения:

n - размерность массива;

x - исходный массив;

i - номер текущего элемента массива;

A, B - границы заданного интервала;

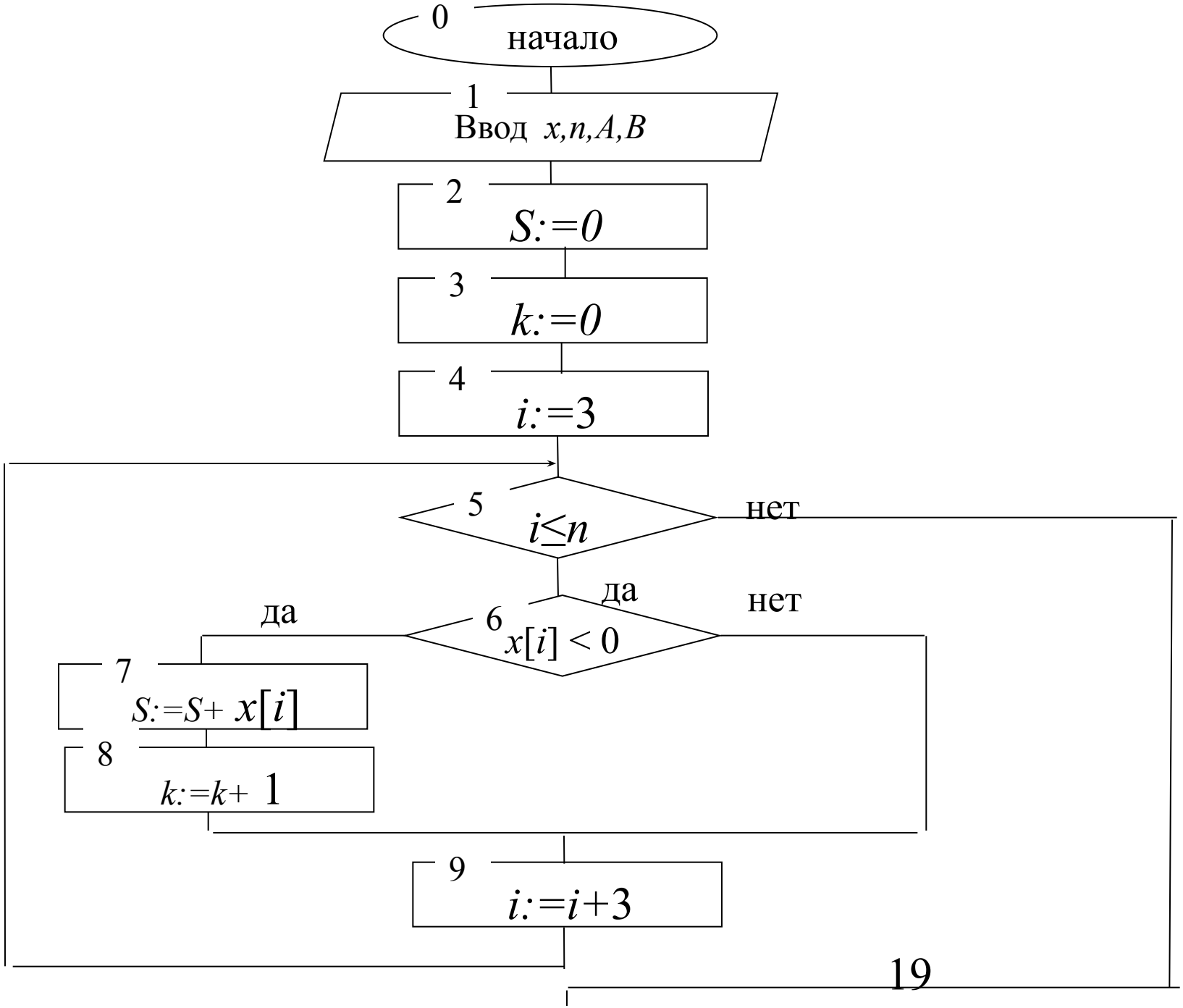
S -сумма отрицательных элементов, стоящих на местах, кратных 3;

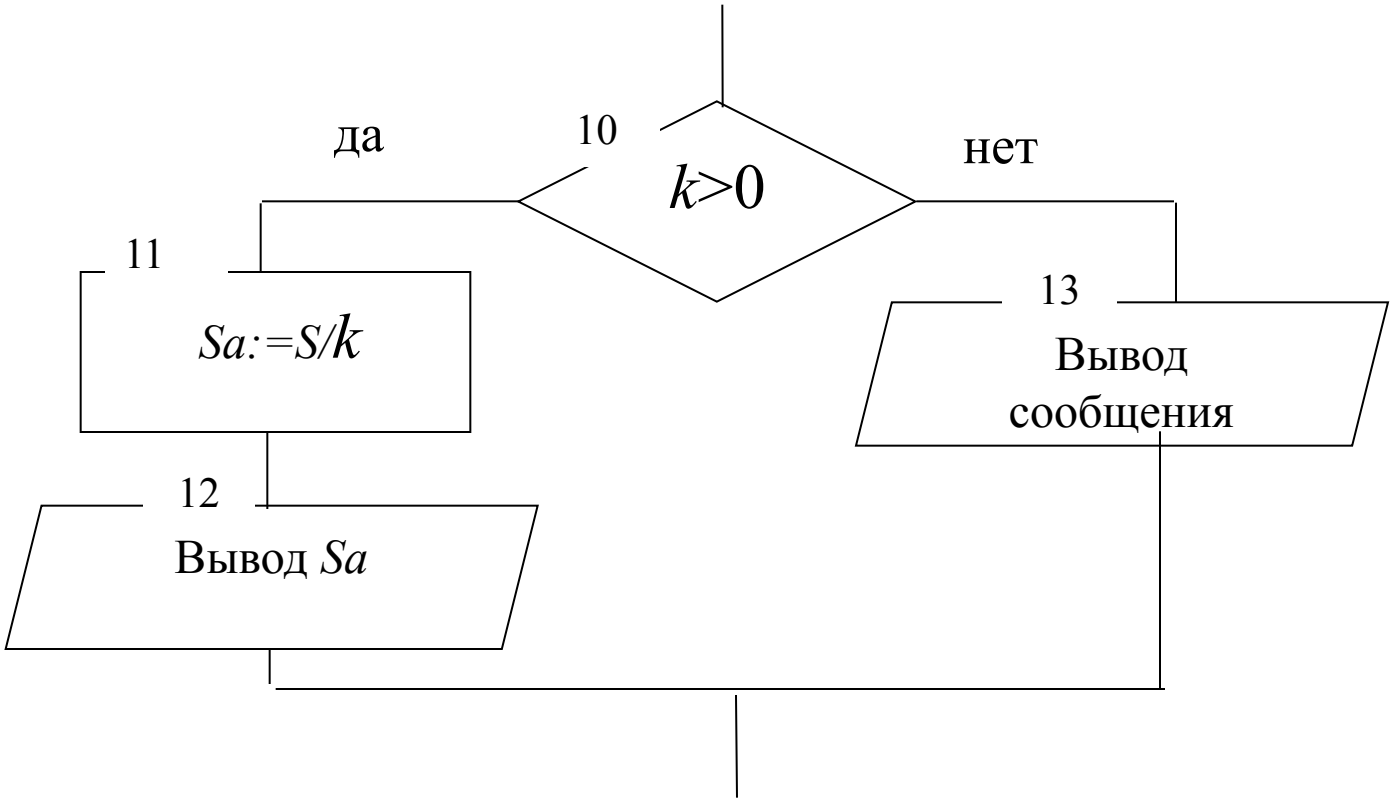
k -количество отрицательных элементов, стоящих на местах, кратных 3;

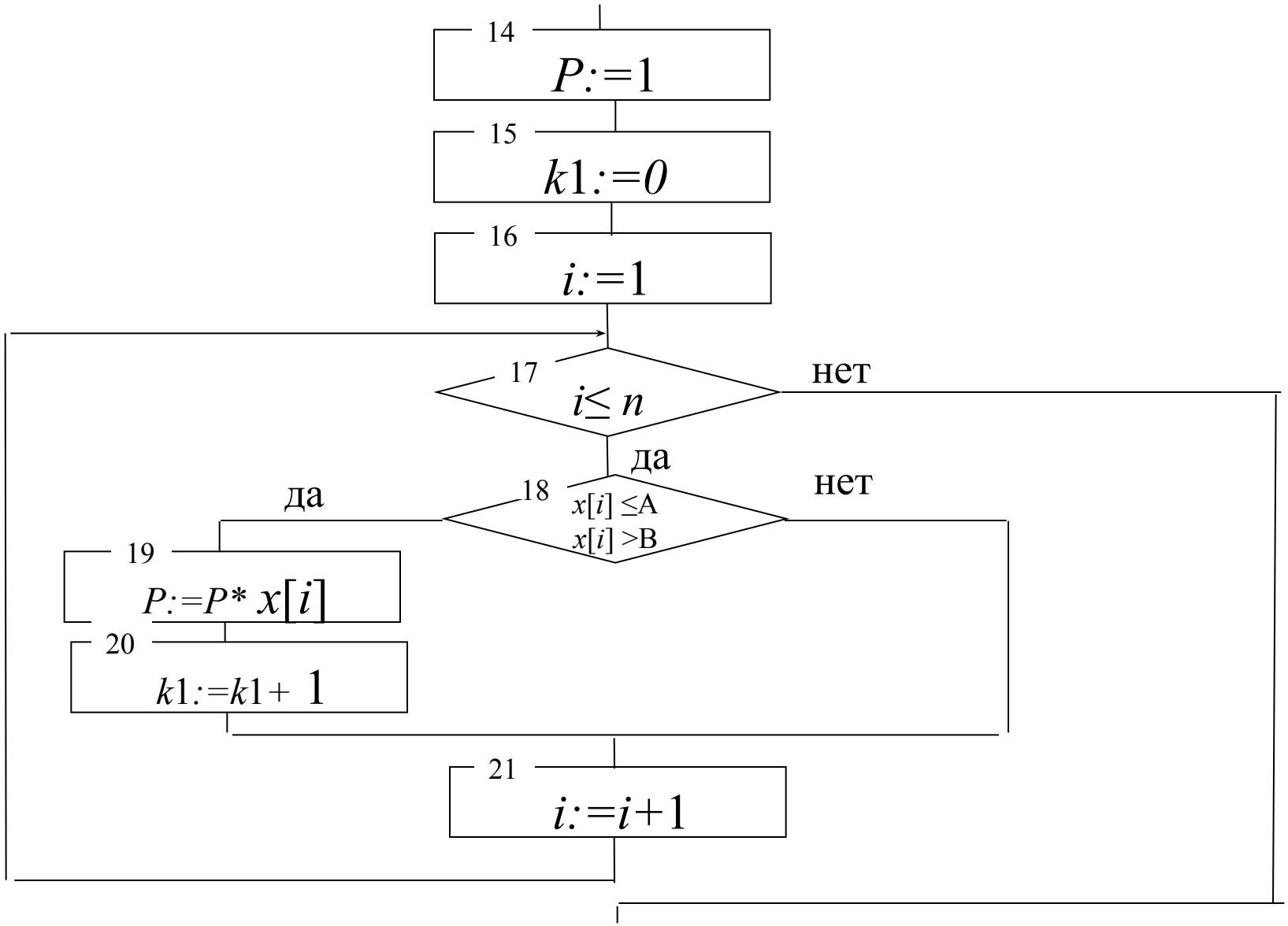
Sa -среднее арифметическое отрицательных элементов, стоящих на местах, кратных 3;

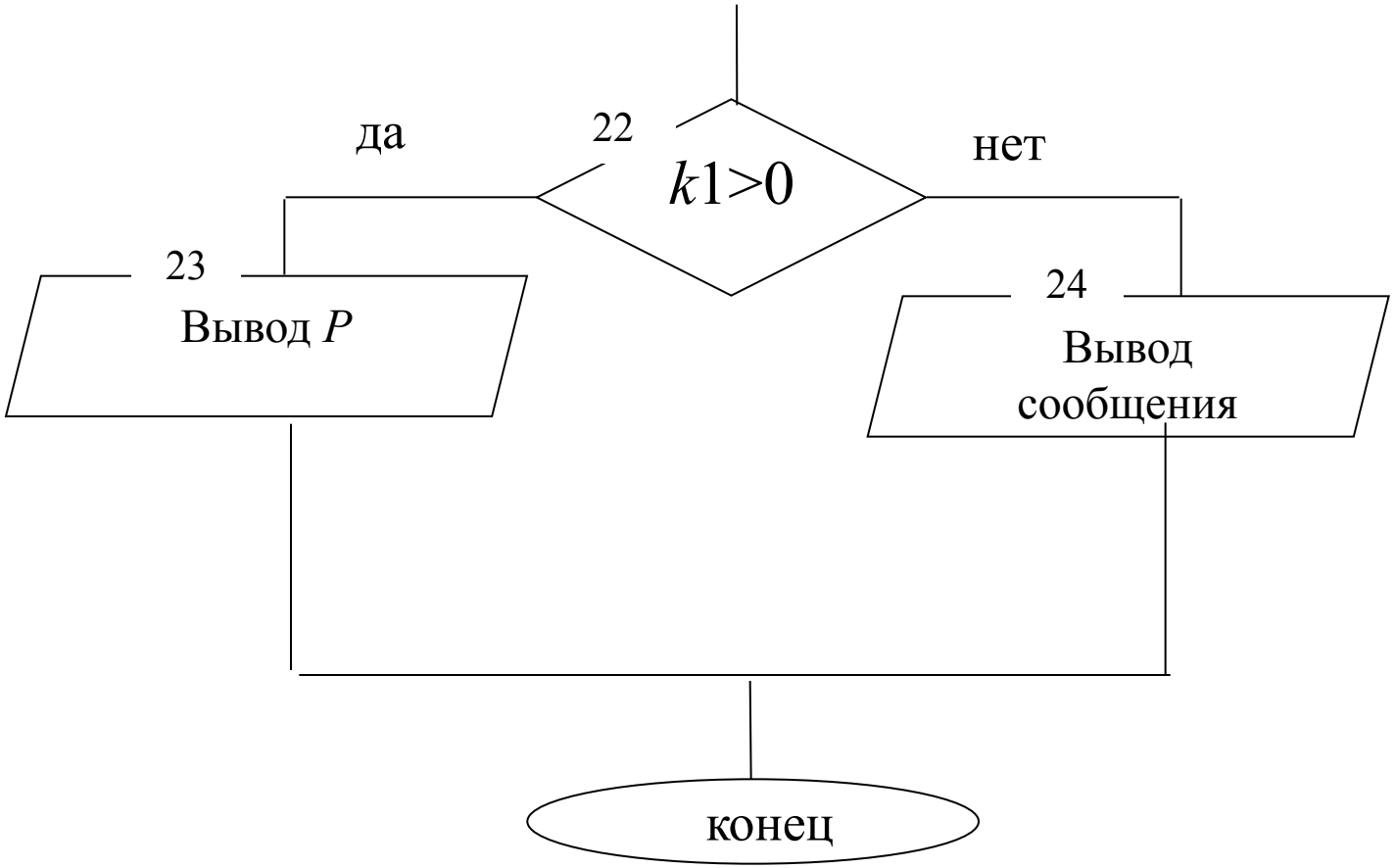
P -произведение элементов, не принадлежащих интервалу (A , B];

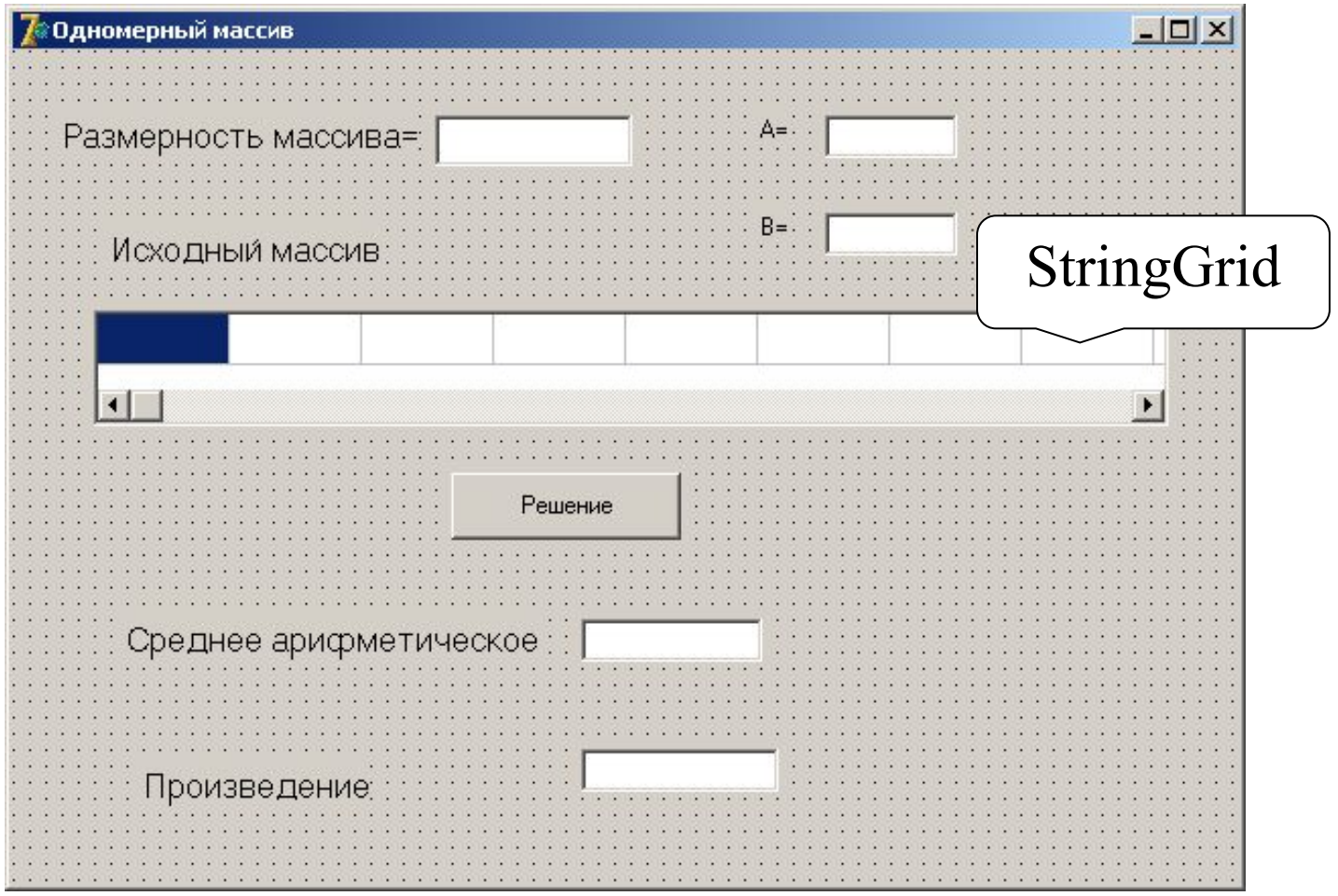
*k*1 –количество элементов, не принадлежащих интервалу (A , B];


0
начало
1
Ввод x,n,A,B
2
S:=0
3
k:=0
4
i:=3
5
i≤n
нет
да
6
x[i] < 0
да
нет
7
S:=S+ x[i]
8
k:=k+ 1
9
i:=i+3


10
k>0
да
нет
11
Sa:=S/k
12
Вывод Sa
13
Вывод
сообщения


14
P:=1
15
k1:=0
16
i:=1
17
i≤ n
нет
да
18
x[i] ≤A
x[i] >B
да
нет
19
P:=P* x[i]
20
k1:=k1+ 1
21
i:=i+1


да
22
нет
k1>0
23
Вывод P
24
Вывод
сообщения
конец


Одномерный массив
Размерность массива=
A=
B=
Исходный массив
StringGrid
Решение
Среднее арифметическое
Произведение

```
procedure TForm1.Button1Click(Sender: TObject);
var
   i,n,k,k1:integer;
   x:array[1..20] of integer;
   S, Sa, P,A,B: real;


BEGIN
  n:=strtoint(edit1.Text);
  for i:=1 to n do
    x[i]:=strtoint(stringgrid1.Cells[i-1,0]);

 A:=StrToFloat(Edit4.Text);  B:=StrToFloat(Edit5.Text);
```

```
S:=0; k:=0;
i:=3;
while i<= n do
  begin
    if  x[i]<0  then  begin
                        S:=S+x[i];
                        k:=k+1
                      end;
     i:=i+3
 end;
```

```
if  k>0 then
      begin
        Sa:=S/k;
        Edit2.Text:=FloatToStr(Sa);
      end
    else
     begin
      Edit2.Text:='Нет!';
      ShowMessage('В массиве нет отрицательных чисел с
                                                  номерами, кр. 3 ');
     end;
```

```
P:=1; k1:=0;
for i:= 1 to n  do
 if (x[i]<=A) or (x[i]>B) then
                    begin
                      P:=P*x[i];
                      k1:=k1+1
                    end;
```

```
if   k1>0  then
    Edit3.Text:=FloatToStr(P)
  else
     begin
       Edit3.Text:='Нет!';
       ShowMessage('Нет элементов, не принадл. интервалу');
     end;
end;
```

Для проверки работы программы понадобятся два набора исходных данных:

1. для проверки работы в случае, когда в массиве нет элементов, удовлетворяющих условию задачи.

Например,

x = (-2 3 3 1 2 2 -1 0 2 2 -2 1)

Заданный интервал (-3, 3].

Результатом должны быть сообщения

'Нет элементов, не принадл. интервалу'

'В массиве нет отрицательных чисел с номерами, кр. 3 '

2. для проверки работы в случае, когда в массиве есть элементы, удовлетворяющие условию задачи.

ShowMessage – стандартная процедура для вывода на экран окна с сообщением и кнопкой ОК.

ShowMessage(<сообщение>);

<сообщение> - выражение строкового типа, например, любая строковая константа – набор символов, заключенных в апострофы.

После появления окна с сообщением работа программы приостанавливается в ожидании реакции пользователя.

Поиск максимального и минимального элементов массива.

Введем следующие обозначения:

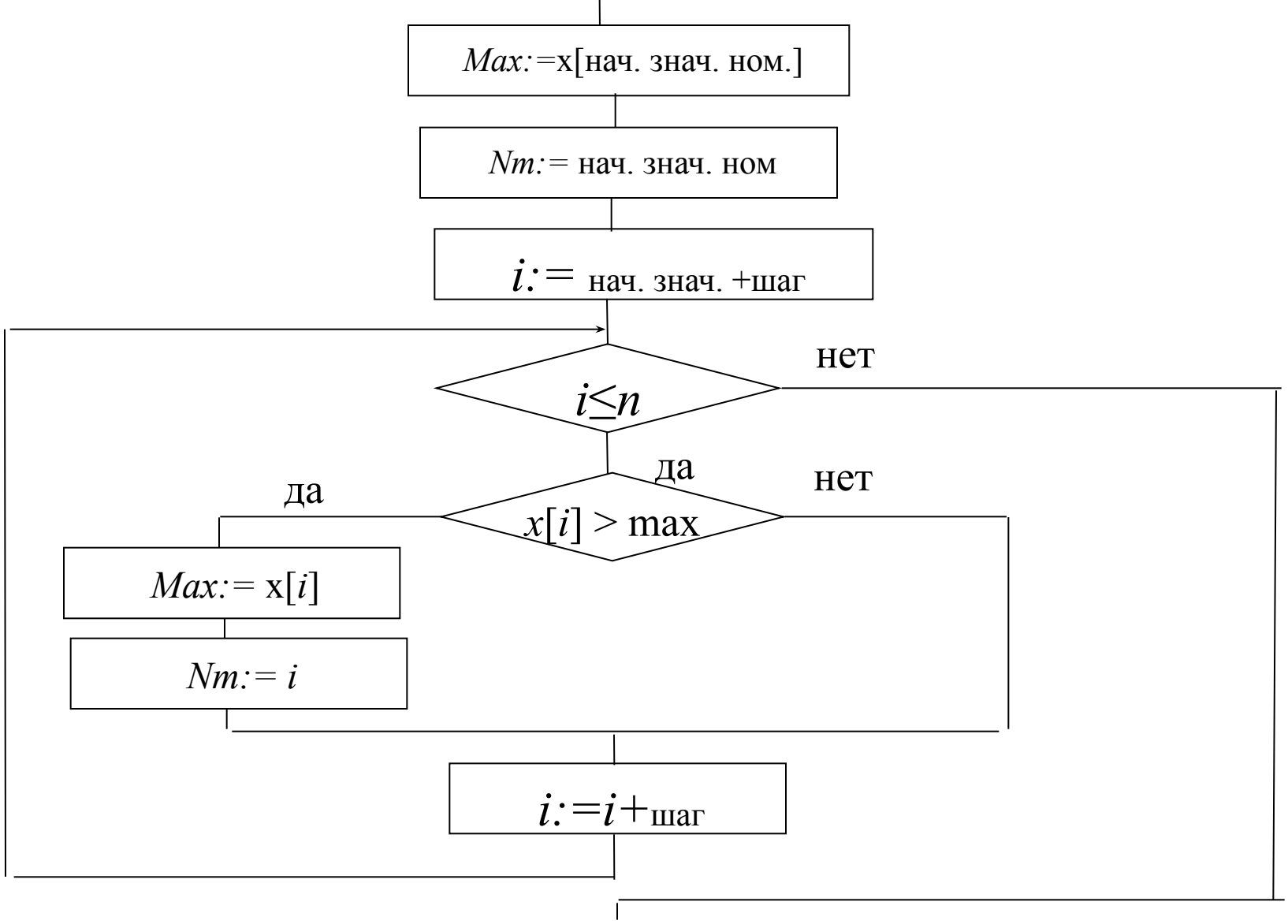
n - размерность массива;

x - исходный массив;

i - номер текущего элемента массива;

Max - значение максимального элемента;

Nm - номер максимального элемента;

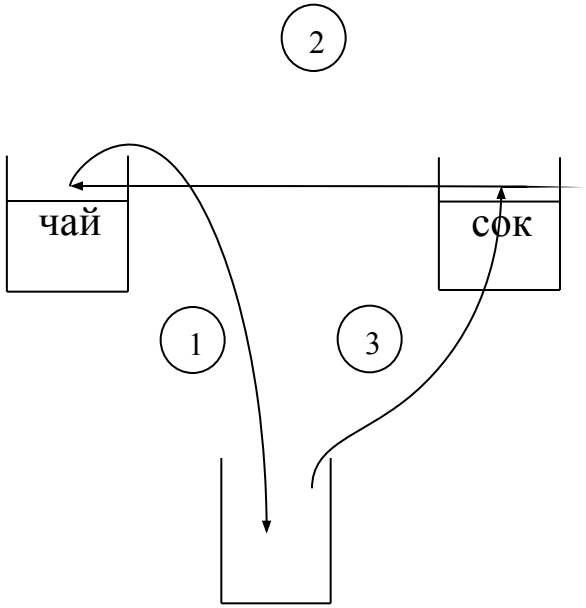


Алгоритм поиска минимального элемента аналогичен.

4.6 Перестановка элементов в массиве.

Чтобы поменять местами два элемента массива, можно применить так называемый метод трех стаканов.

Суть его заключается в следующем: *чтобы поменять местами жидкости в двух стаканах, нужен третий пустой стакан*.



Для перестановки элементов массива также требуется дополнительная переменная.

Тип этой переменной должен совпадать с типом элементов массива.

Пусть имеется следующее описание переменных

```
var
   k1,k2:integer;
    x:array[1..30] of real;
   z:real;
```

Тогда обмен местами элементов массива с номерами k1 и k2 в программе можно осуществить след. образом:

```
z:=x[k1];
x[k1]:=x[k2];
x[k2]:=z;
```

Пример. В заданном массиве найти максимальный элемент и поменять его местами с предыдущим элементом.

Например, пусть задан массив

x = (-2 3 -3 1 4 -5 -1 0 2 8 -3 -7)

Тогда максимальный элемент равен **8**.

Массив после обмена должен принять следующий вид:

x = (-2 3 -3 1 4 -5 -1 0 8 2 -3 -7)

Нахождение максимального и минимального элементов массива.

Пример 2. В заданном массиве найти максимальный элемент и поменять его местами с предыдущим элементом.

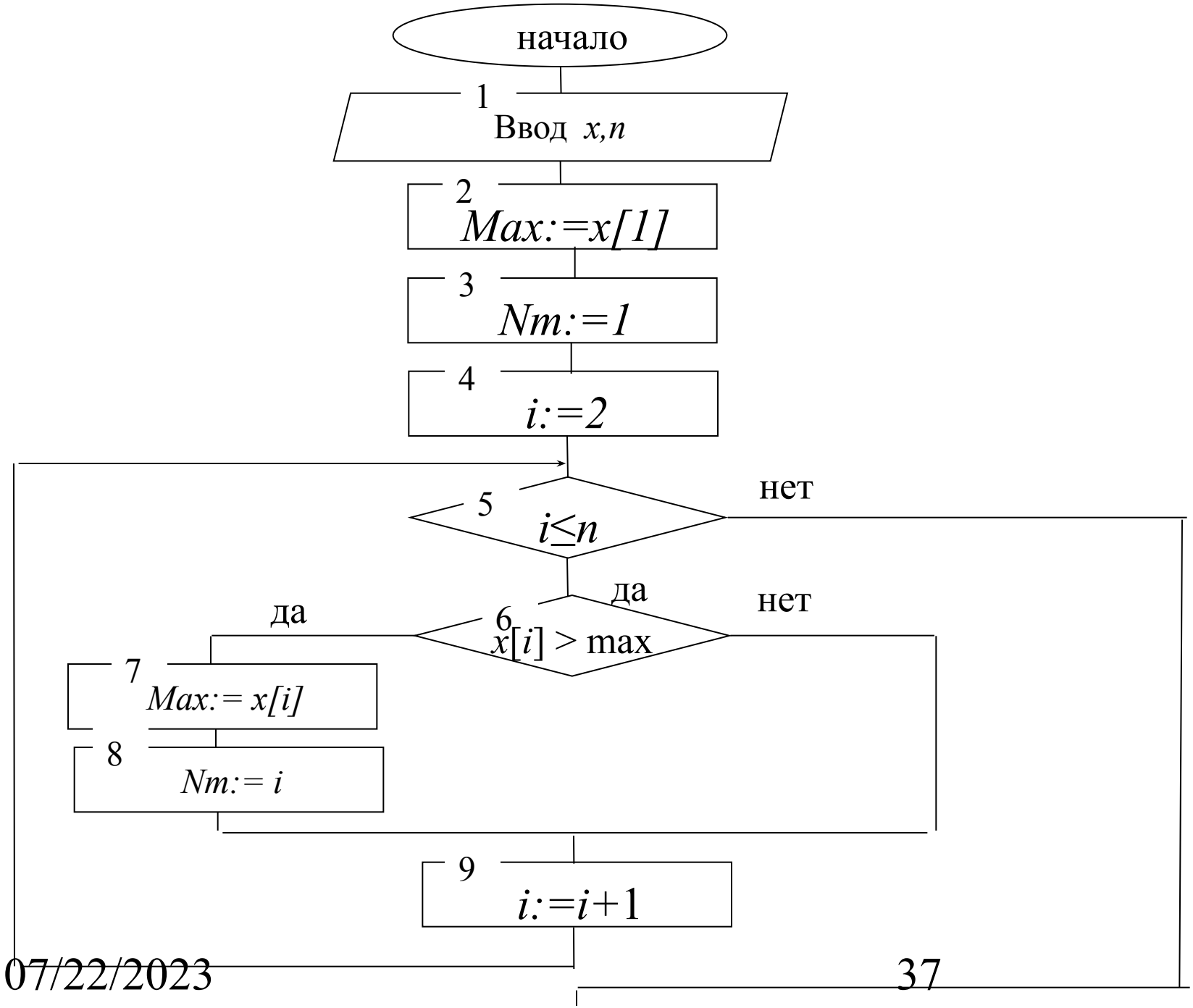
Например, пусть задан массив

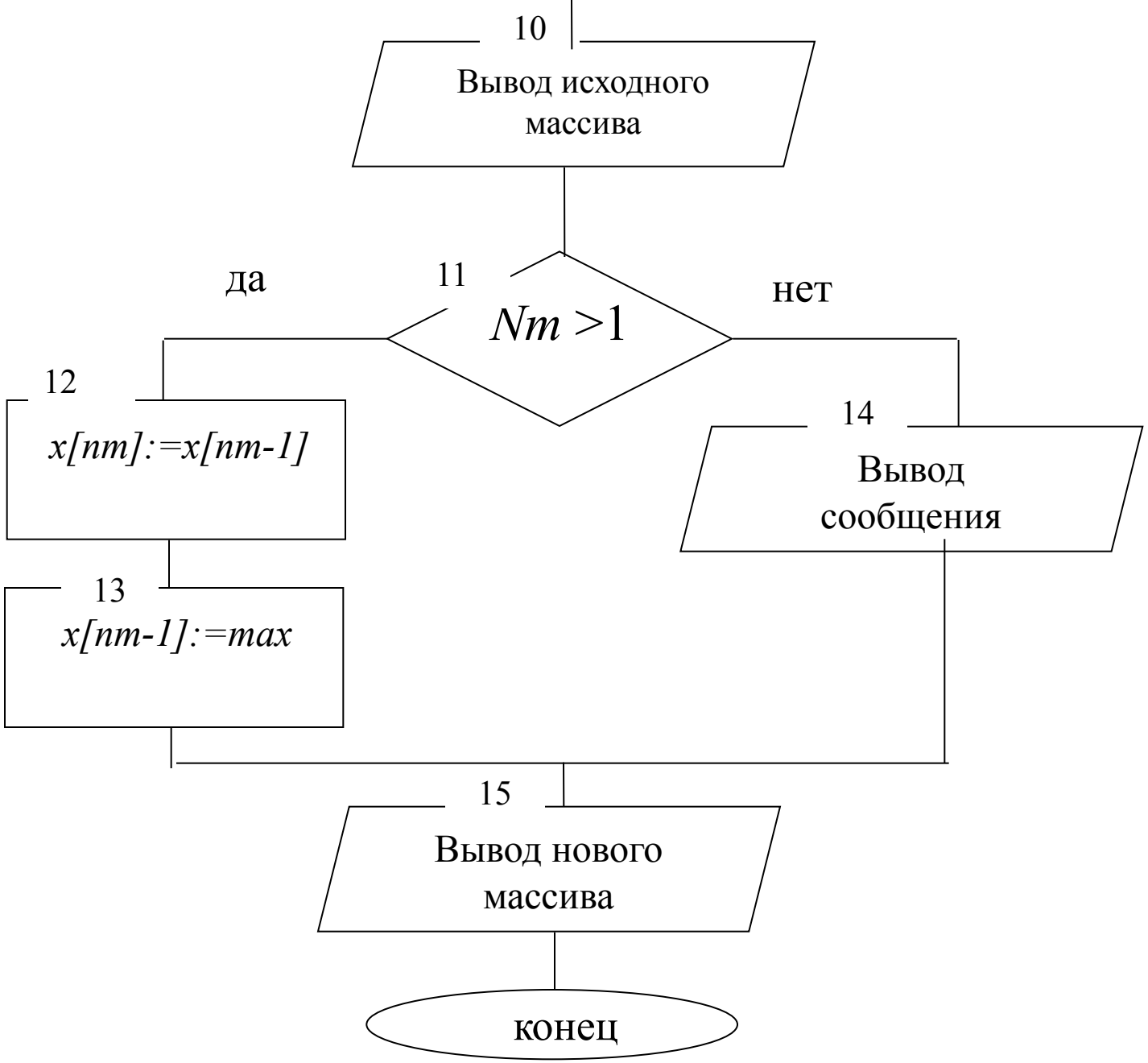
x = (-2 3 -3 1 4 -5 -1 0 2 8 -3 -7)

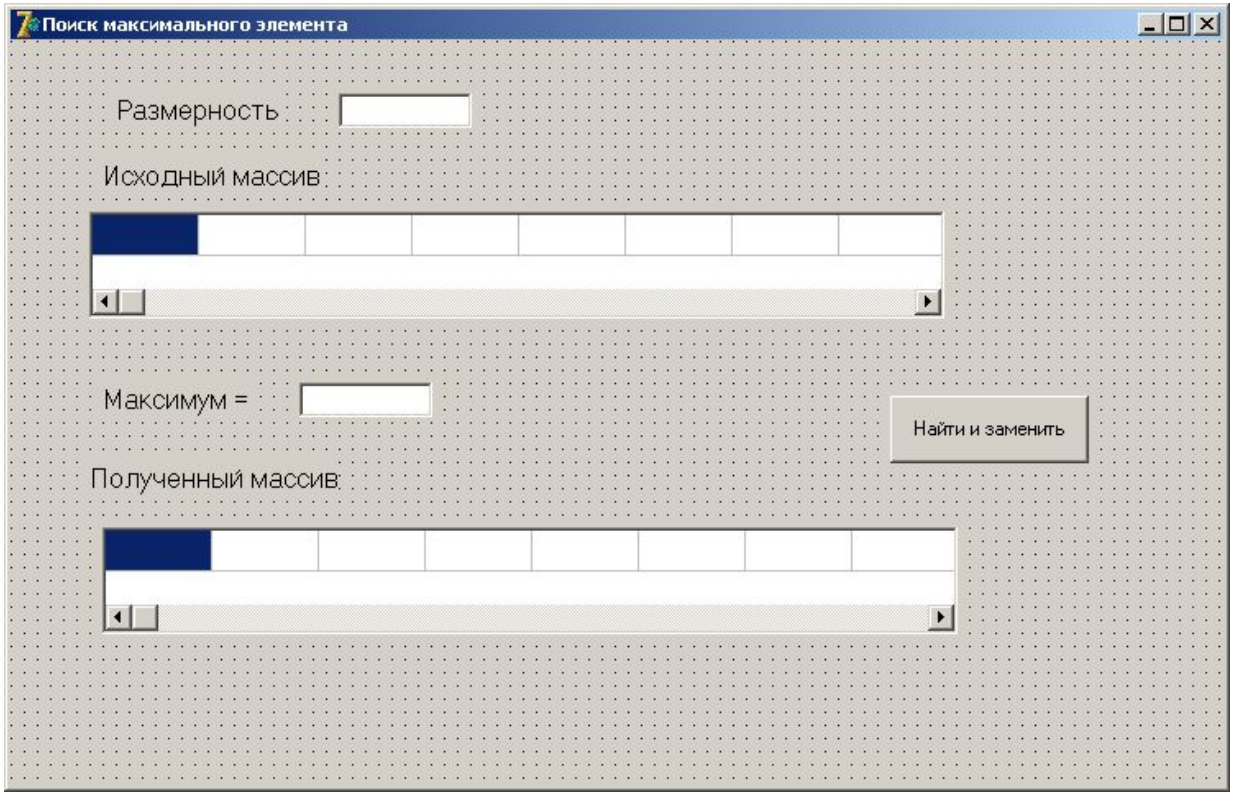
Тогда максимальный элемент равен **8**.

Массив после обмена должен принять следующий вид:

x = (-2 3 -3 1 4 -5 -1 0 8 2 -3 -7)


начало
1
Ввод x,n
2
Max:=x[1]
3
Nm:=1
4
i:=2
5
i≤n
нет
да
6
x[i] > max
да
нет
7
Max:= x[i]
8
Nm:= i
9
i:=i+1


10
Вывод исходного массива
11
Nm >1
да
нет
12
x[nm]:=x[nm-1]
13
x[nm-1]:=max
14
Вывод сообщения
15
Вывод нового массива
конец


Поиск максимального элемента
Размерность
Исходный массив
Максимум =
Найти и заменить
Полученный массив

```
Procedure TForm1.Button1Click(Sender:
TObject);
 var
    i,n,max,nm:integer;
    x:array[1..20] of integer;

 begin
    n:=strtoint(edit1.Text);
    for i:=1 to n do
      x[i]:=strtoint(stringgrid1.Cells[i-1,0]);

    max:=x[1];
    nm:=1;

    for i:=2 to n do
     if x[i]>max then
         begin
           max:=x[i]; nm:=i
         end;
```

```
  if nm<>1 then
     begin
      x[nm]:=x[nm-1];
      x[nm-1]:=max
     end
    else
    ShowMessage('Максимальный элемент стоит на первом месте!');


 edit2.Text:=inttostr(max);
    for i:=1 to n do
      stringgrid2.Cells[i-1,0]:=Inttostr(x[i]);
end;
```

5. Создание приложений с несколькими формами

Чтобы добавить к приложению новую форму, нужно выполнить команду главного меню **File – New Form** или нажать кнопку **New Form** на панели инструментов. 

В результате к приложению будет добавлена пустая форма **Form2 (**или **Form3, Form4 и т.д.)** и соответствующий ей модуль **Unit 2.**

При этом в файл проекта будут автоматически добавлены следующие строки программы:

`Unit2 in 'Unit2.pas' {Form2}` в разделе подключения модулей **Uses**

и

```
Application.CreateForm(TForm2, Form2);
```

{ создание формы} в разделе операторов.

Та форма, которая в файле проекта создается первой, является главной, и окно этой формы появляется на экране при запуске программы на выполнение.

По умолчанию это форма, которая была автоматически создана при создании приложения (**Form1)**.

Если необходимо, чтобы в момент старта программы появлялось окно любой другой формы, нужно в файле проекта оператор создания этой формы

Application.CreateForm(<имя класса>,<имя формы>);

расположить раньше операторов создания всех остальных форм.

Сделать форму главной можно также, воспользовавшись командой меню **Project – Options** и на вкладке **Forms** выбрав в списке **Main Form** нужную форму**.**

Окна всех остальных форм появляются на экране только после обращения к методам **Show** или **ShowModal** в тексте программы:

<имя формы>. Show;
<имя формы>. ShowModal;

Например, **Form2. ShowModal;**

Метод **Show** показывает форму в немодальном режиме, т.е. окно появляется на экране и работает одновременно с другими окнами. При этом управление сразу передается оператору, стоящему за обращением к этому методу.

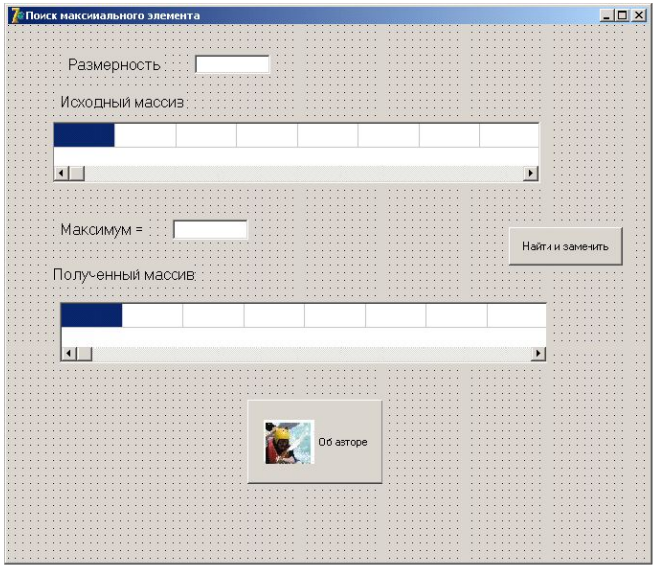
Метод **ShowModal** показывает форму в модальном режиме, т.е. доступным будет окно только этой формы, а другие окна станут доступными только после закрытия модального окна. При этом оператор программы, следующий за обращением к методу **ShowModal,** получит управление также только после закрытия этого окна.

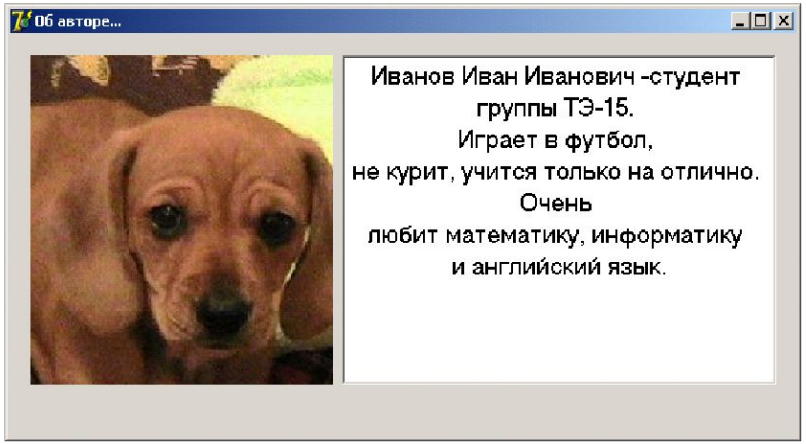
Чтобы получить доступ к методам и визуальным компонентам формы из модуля, в котором эта форма не описана, нужно подключить к этому модулю в разделе **Uses** модуль, соответствующий форме.

Это можно сделать следующим образом:

- Набрать в разделе **implementation** предложение
  Uses <имя модуля>;
- Выполнить команду меню
  File – Uses Unit
  и выбрать в появившемся окне нужный модуль.
- Попытаться запустить программу на выполнение и в ответ на сообщение Delphi нажать кнопку **Yes**.

Например, добавим в программу из предыдущего примера окно с информацией об авторе, которое будет появляться после нажатия на соответствующую кнопку.





Для этого на главной форме разместим кнопку **BitBtn** и установим для нее все необходимые свойства: **Caption** – Об авторе, **Glyph –** файл с фотографией.

Создадим новую форму и разместим на ней компонент **Memo**. Установим для него все необходимые свойства: шрифт, размер, выравнивание, в том числе в свойстве **Lines** наберем нужный текст.

Разместим на форме компонент **Image (**страница **Additional)** для вставки фотографии**.** Используя свойство **Picture** загрузим соответствующий графический файл.

Подключим модуль **Unit2** к модулю **Unit1**. Для этого активизируем главную форму и выполним команду меню **File – Uses Unit.**

В результате в реализационной части (implementation) модуля Unit1 появится предложение

```
uses Unit2;
```

Создадим процедуру обработки события нажатия кнопки **Об авторе** и обратимся в разделе операторов этой процедуры к методу Show или ShowModal второй формы:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
 Form2.Show
end;
```

Если нужно вывести результаты вычислений в новом окне, нужно создать новую форму и разместить на ней необходимые для вывода визуальные компоненты.

В тексте программы при выводе результатов при обращении к визуальным компонентам новой формы нужно перед именем ВК указывать имя формы.

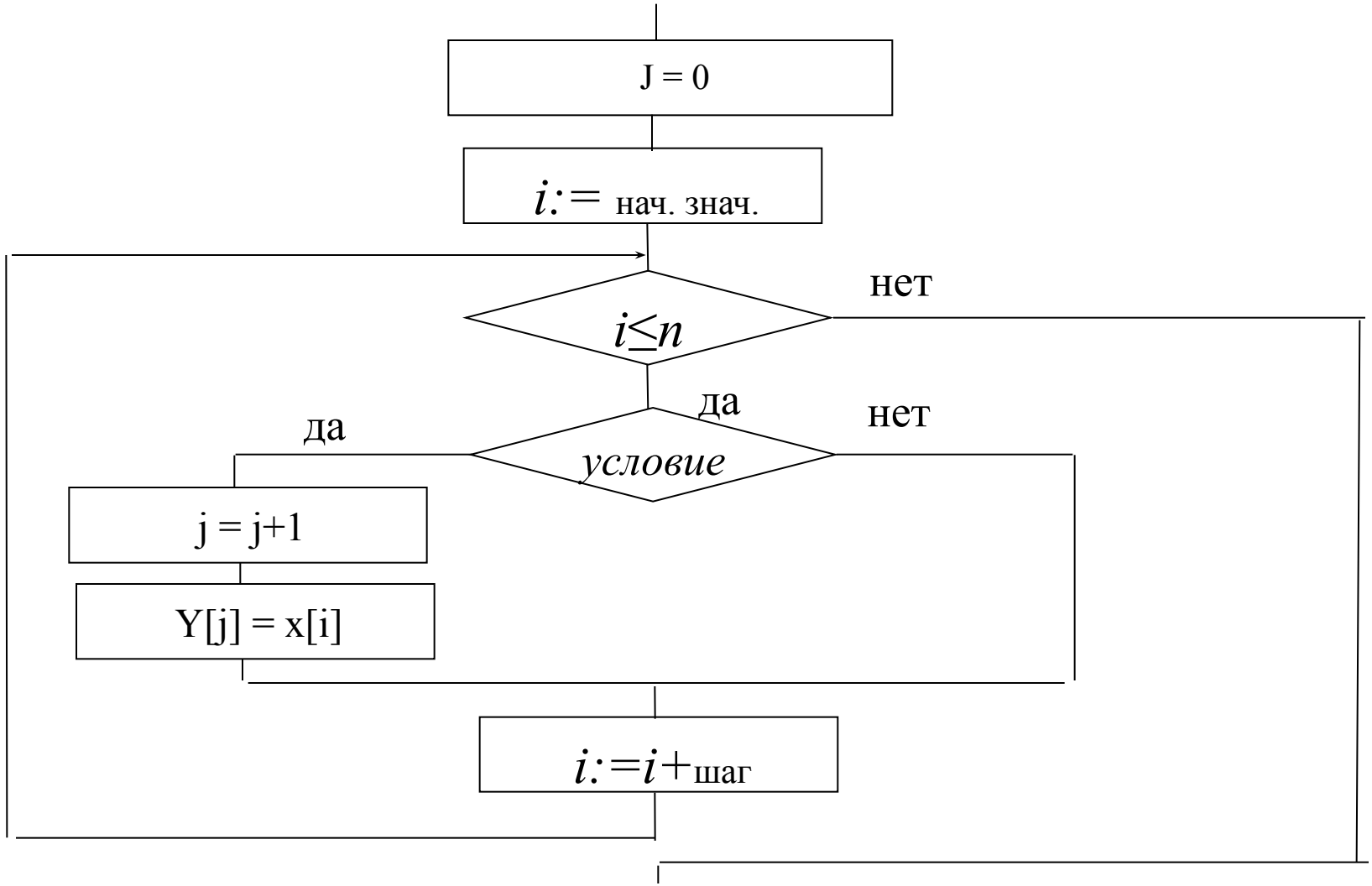
Например, для вывода значения вещественной переменной *x* на форме **Form2** нужно набрать

```
Form2.Edit1.Text:=FloatToStr(x);
```

6. Формирование нового массива из элементов исходного, удовлетворяющих заданному условию.

Пусть *n* - размерность массива; *x* - исходный массив;

i - номер текущего элемента массива, j - количество элементов нового массива, y – новый массив.


J = 0
i:= нач. знач.
i≤n
нет
да
нет
да
условие
j = j+1
Y[j] = x[i]
i:=i+шаг

```
J:=0;
  i:=<нач. Знач.>;
   while i<=n do
    begin
     if <условие> then
       begin
         j:=j+1;
         y[j]:=x[i]
       end;
       i:=i+<шаг>
     end;
```

Пример. Имеется список студентов МСФ, для каждого студента известен рост. Составить список студентов выше 180 см для формирования баскетбольной команды.

Обозначим

St – одномерный массив из строк, элементами которого являются фамилии студентов.

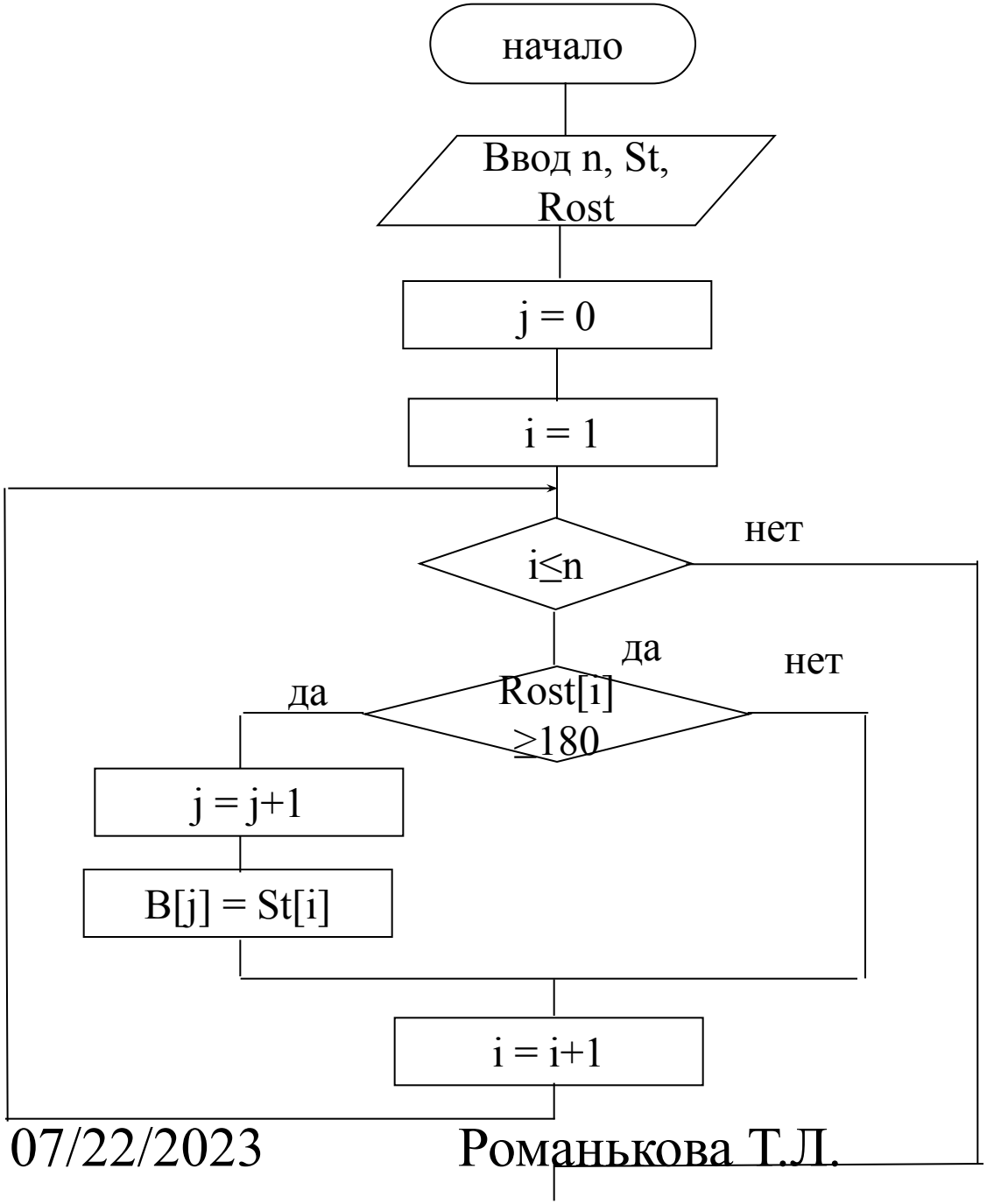
Rost - одномерный массив целых чисел, элементами которого являются соответствующие ростовые показатели студентов.

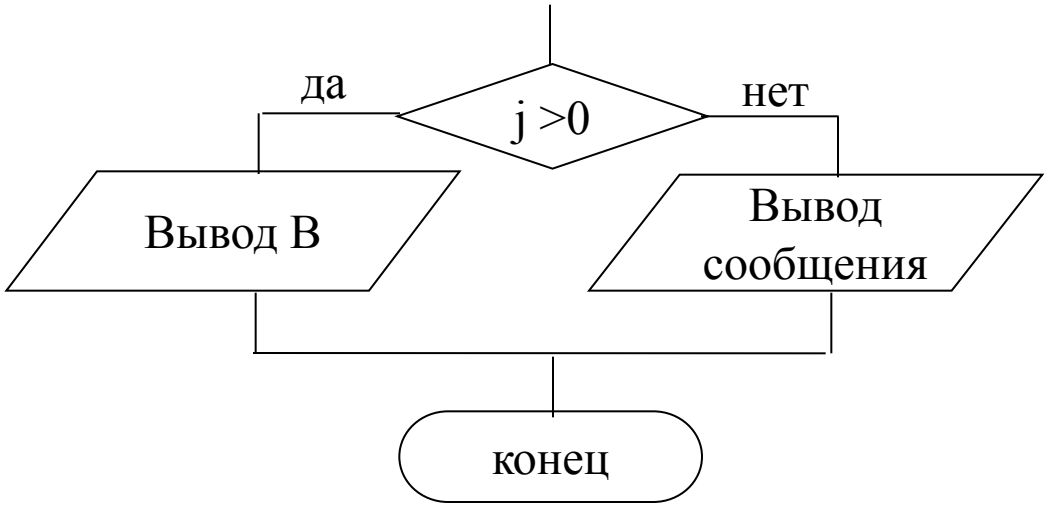
N – количество студентов.

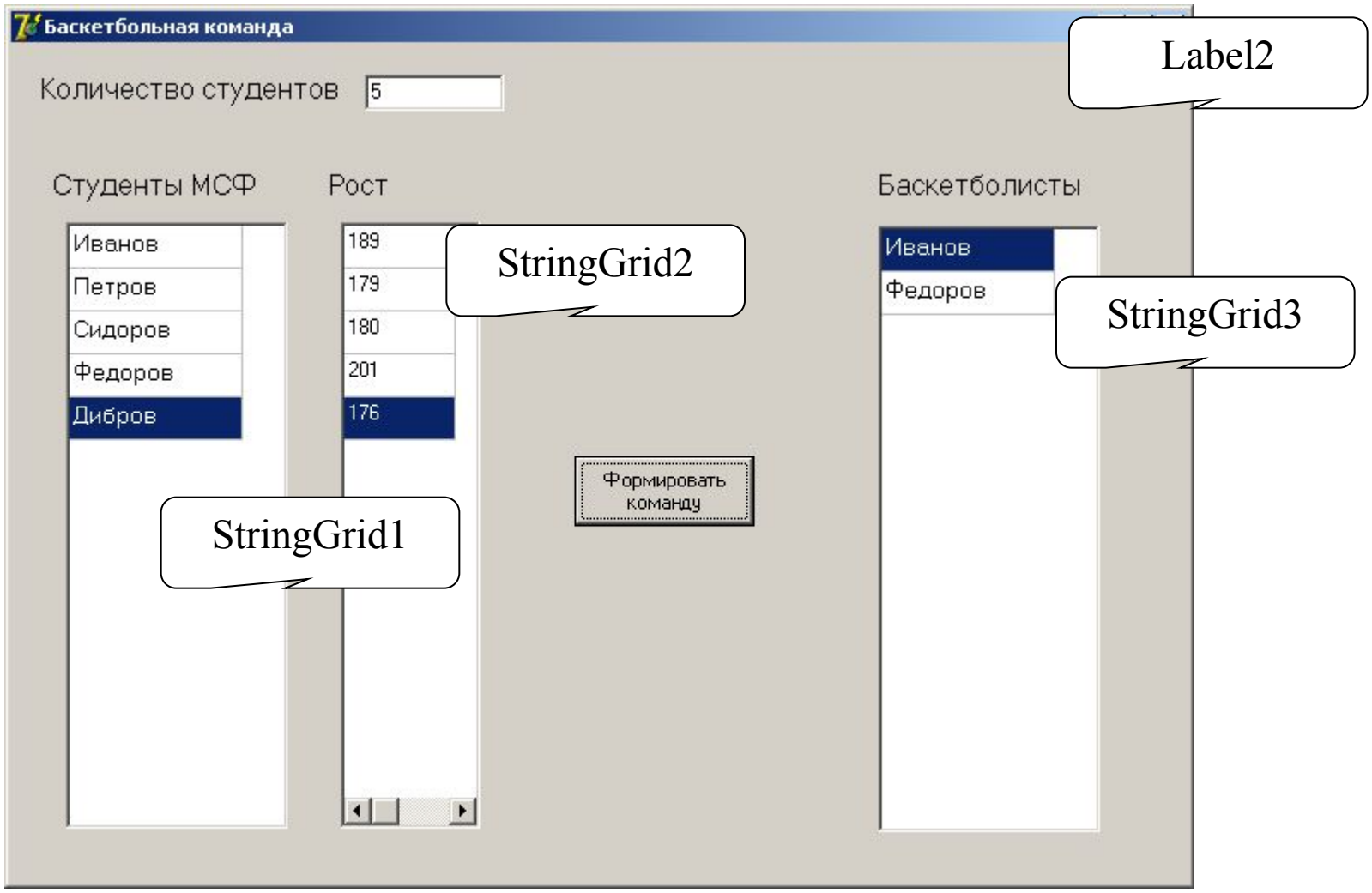
B – одномерный массив строк, содержащий фамилии студентов из баскетбольной команды.

I – номер студента в списке.

J – количество студентов в баскетбольной команде.


начало
Ввод n, St, Rost
j = 0
i = 1
i≤n
нет
да
Rost[i] ≥180
да
нет
j = j+1
B[j] = St[i]
i = i+1


да
j >0
нет
Вывод B
Вывод
сообщения
конец


Баскетбольная команда
Количество студентов
5
Студенты МСФ
Рост
Баскетболисты
Иванов
Петров
Сидоров
Федоров
Дибров
189
179
180
201
176
Иванов
Федоров
Формировать команду
Label2
StringGrid2
StringGrid3
StringGrid1

Для компонентов Label2 и StringGrid3 установим для свойства Visible значение False, чтобы до нажатия кнопки они не отражались в окне программы.

Чтобы при изменении количества студентов автоматически изменялось бы количество строк в таблицах StringGrid1 и StringGrid2, создадим процедуру:

```
procedure TForm1.Edit1Change(Sender: TObject);
begin
if  edit1.text<>'' then
 Begin
   stringgrid1.RowCount:=StrToInt(edit1.text);
   stringgrid2.RowCount:=StrToInt(edit1.text)
 end
end;
```

Текст процедуры обработки события нажатия кнопки:

```
procedure TForm1.Button1Click(Sender: TObject);
var
   i,n,j:integer;
   Rost:array[1..20] of integer;
   St,B:array[1..20] of string;


Begin
{Ввод исходных данных}
   n:=strtoint(edit1.Text);
   for i:=1 to n do
     st[i]:=stringgrid1.Cells[0,i-1];
   for i:=1 to n do
     rost[i]:=strtoint(stringgrid2.Cells[0,i-1]);
```

```
   {Формирование нового массива}
   j:=0;
   for i:=1 to n do
    if Rost[i]>180 then
        begin
          j:=j+1;
          B[j]:=st[i]
         end;
{Вывод результата}
if j>0 then
   begin
    label2.Visible:=true;stringgrid3.Visible:=true;
    stringgrid3.RowCount:= j;
     for i:=1 to j do
     stringgrid3.Cells[0,i-1]:=b[i];
   end
 else
  ShowMessage('На МСФ нет студентов для баскетб. команды!');
end;
```

Пример. Заданы два одномерных массива целых чисел**.** Сформировать новый массив из элементов первого массива, превышающих значение минимального элемента второго массива, и элементов второго массива с четными номерами.

Введем следующие обозначения:

*n*1 - размерность 1-го массива;

a – 1-й исходный массив;

*n*2 - размерность 2-го массива;

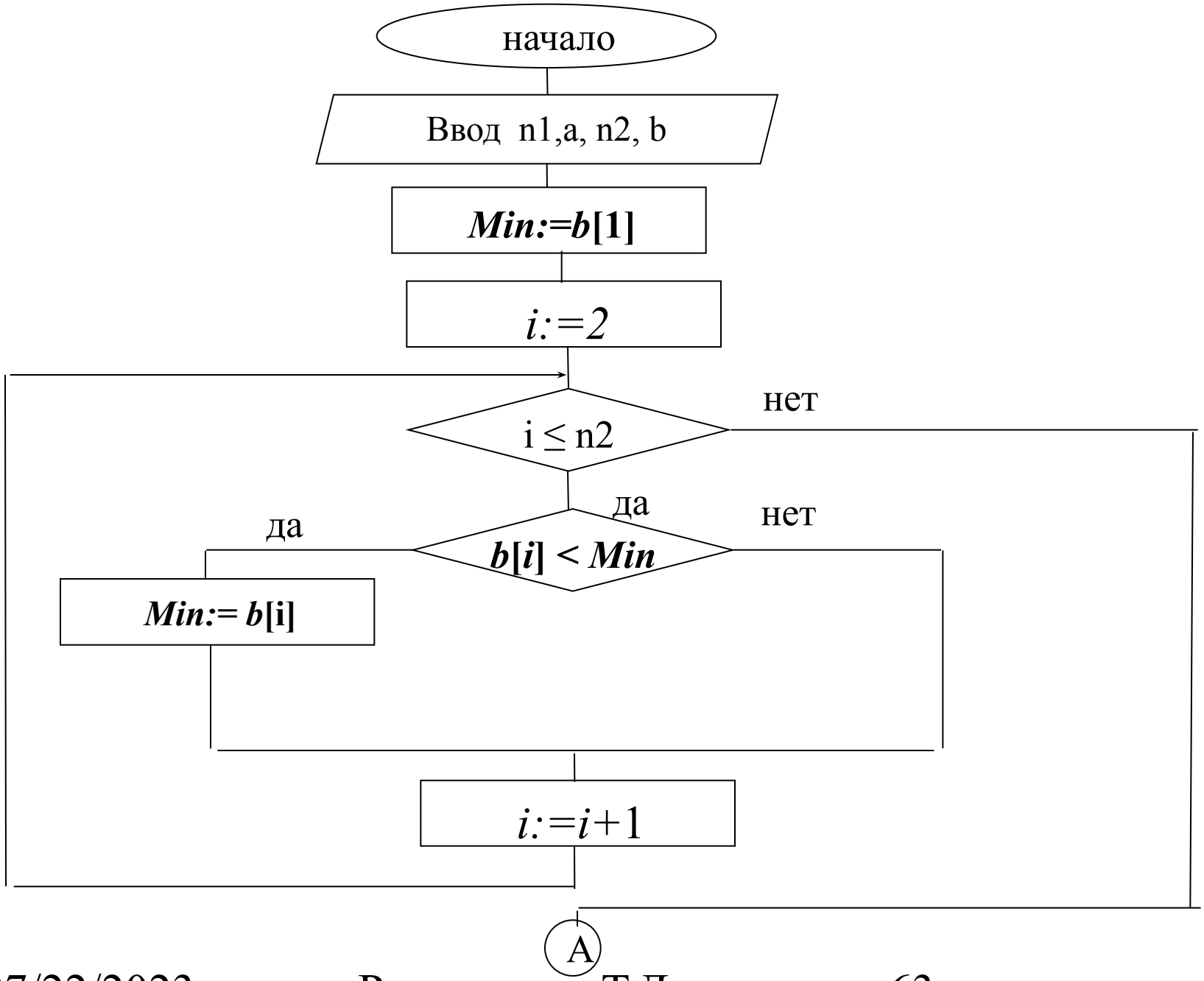
b – 2-й исходный массив;

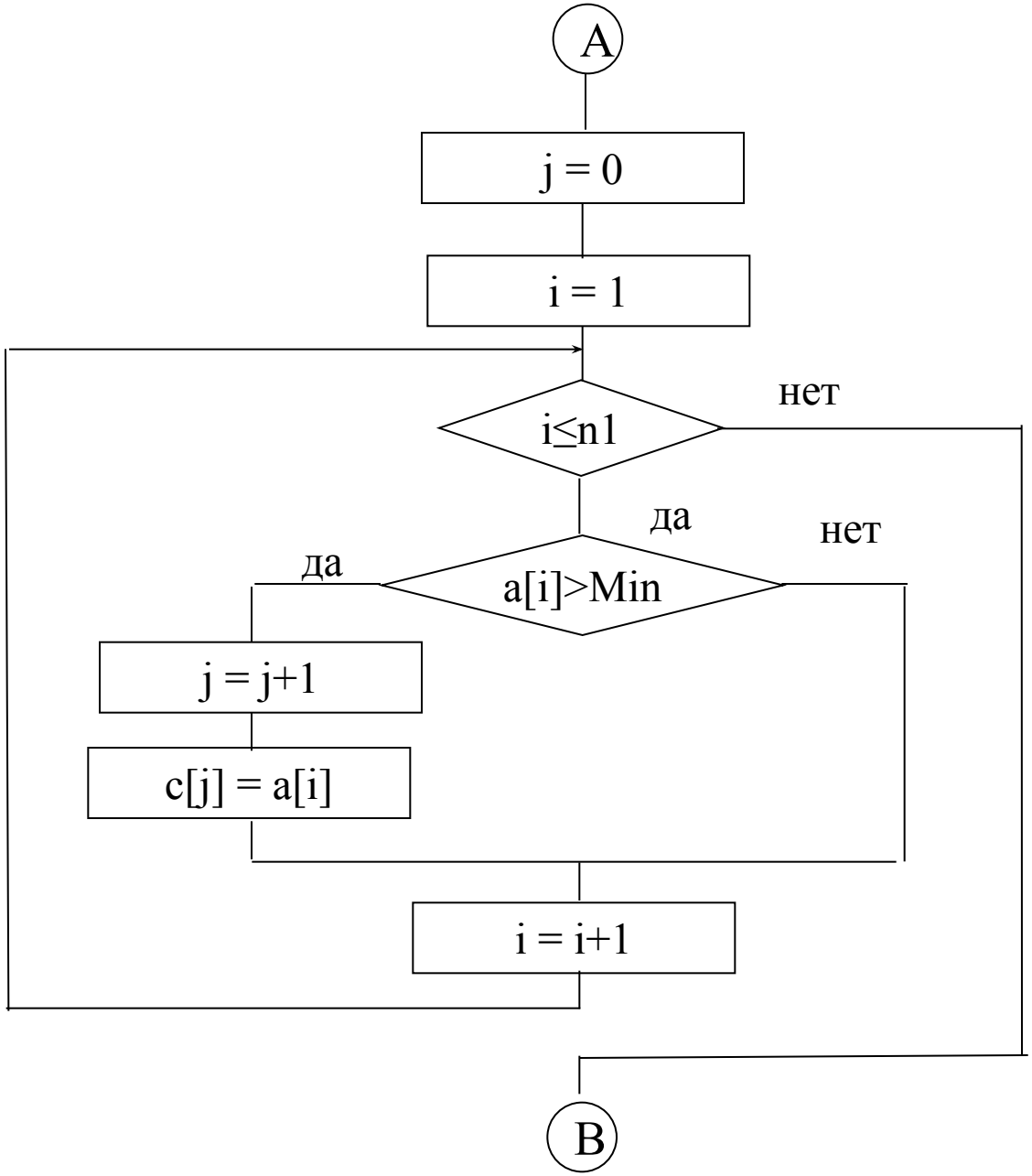
i - номер текущего элемента массива;

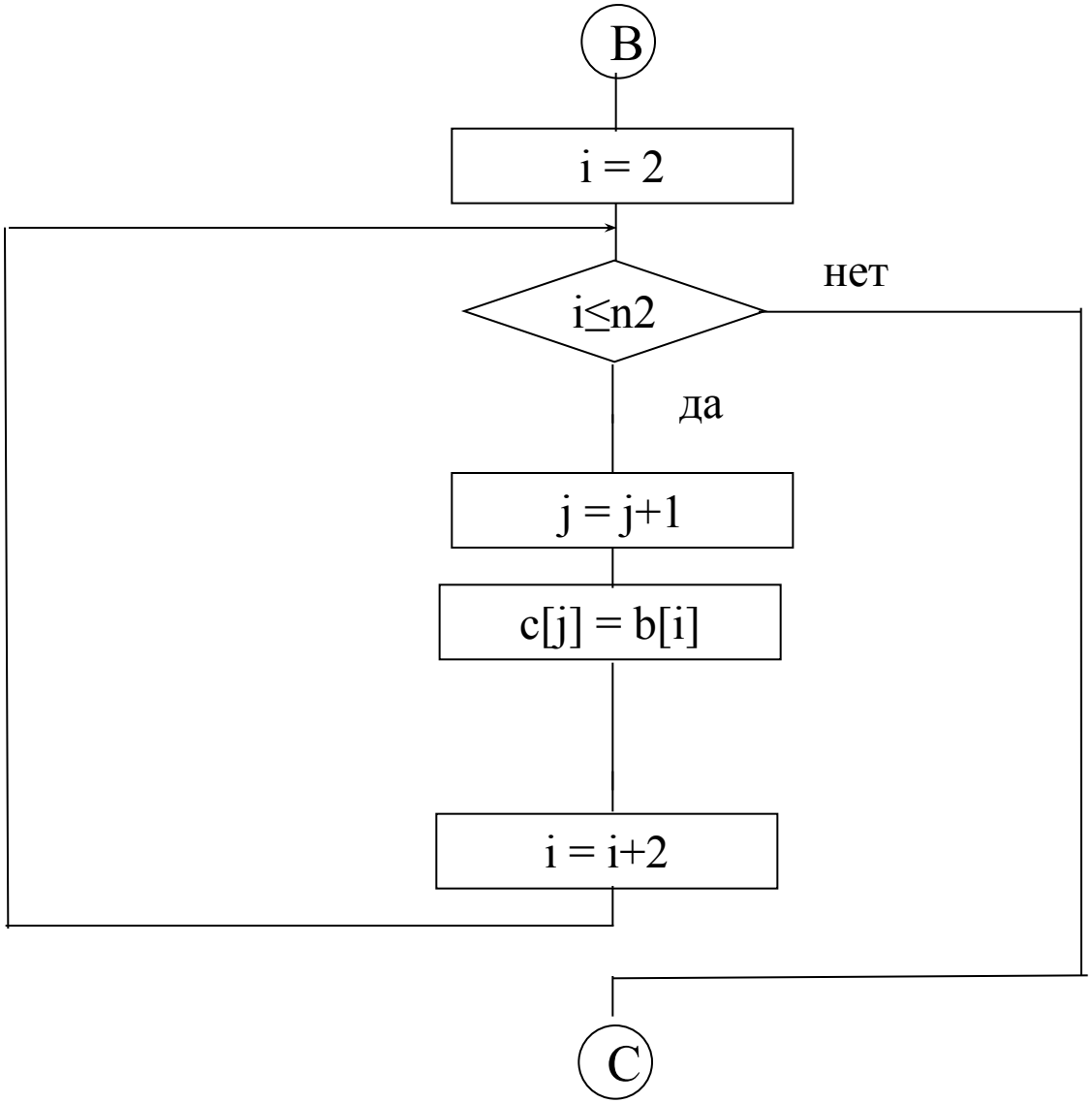
Min - значение минимального элемента 2-го массива;

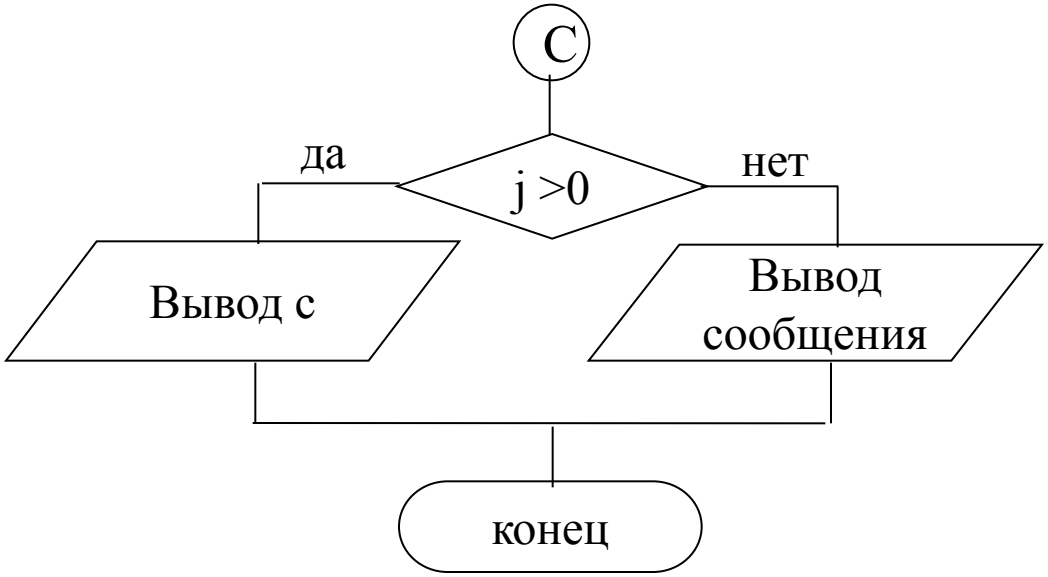
j - размерность нового массива;

c – полученный массив;


начало
Ввод n1,a, n2, b
Min:=b[1]
i:=2
i ≤ n2
нет
да
b[i] < Min
да
нет
Min:= b[i]
i:=i+1
A


A
j = 0
i = 1
i≤n1
нет
да
нет
да
a[i]>Min
j = j+1
c[j] = a[i]
i = i+1
B


B
i = 2
i≤n2
нет
да
j = j+1
c[j] = b[i]
i = i+2
C


C
да
j >0
нет
Вывод с
Вывод
сообщения
конец

```
procedure TForm1.Button1Click(Sender:
TObject);
var
   i,n1,n2,j,Min:integer;
   a,b:array[1..20] of integer;
   c:array[1..40] of integer;

begin
  n1:=strtoint(edit1.Text);
  n2:=strtoint(edit2.Text);
   for i:=1 to n1 do
     a[i]:=strtoint(stringgrid1.Cells[0,i-1]);

   for i:=1 to n2 do
     b[i]:=strtoint(stringgrid2.Cells[0,i-1]);
```

```
min:=b[1];
     for i:=2 to n2 do
      if b[i]< min then
             min:=b[i];


j:=0;
for i:=1 to n1 do
     if a[i]>min then
         begin
           j:=j+1;
           c[j]:=a[i]
          end;
```

```
i:=2;
 while i<=n2 do
    begin
      j:=j+1;
      c[j]:=b[i];
      i:=i+2
     end;

if j>0 then
   begin
     label2.Visible:=true;
     stringgrid3.Visible:=true;
     stringgrid3.RowCount:= j;
     for i:=1 to j do
    stringgrid3.Cells[0,i-1]:=IntToStr(c[i]);
   end
```

```
else
     ShowMessage('Массив не сформирован !')

end;
```

Фокус ввода

На форме может располагаться несколько компонентов, которые могут использовать клавиатуру. Например, Edit, Memo, Button, StringGrid.

Чтобы передать компоненту клавиатурный ввод, нужно передать ему фокус ввода с помощью метода

SetFocus

Например, чтобы при запуске программы на выполнение фокус ввода был у компонента Edit1 , нужно использовать метод SetFocus в процедуре – обработчике события активирования формы:

```
procedure TForm1.FormActivate(Sender:
TObject);
begin
 edit1.SetFocus
end;
```