

## **Лабораторная работа №11**

### **Комбинированные вычислительные процессы.**

#### **2. Цель лабораторной работы:**

Изучить итерационные комбинированные ВП.

#### **3. Используемое оборудование:**

ПК, среда программирования Lazarus.

#### **Задача 1**

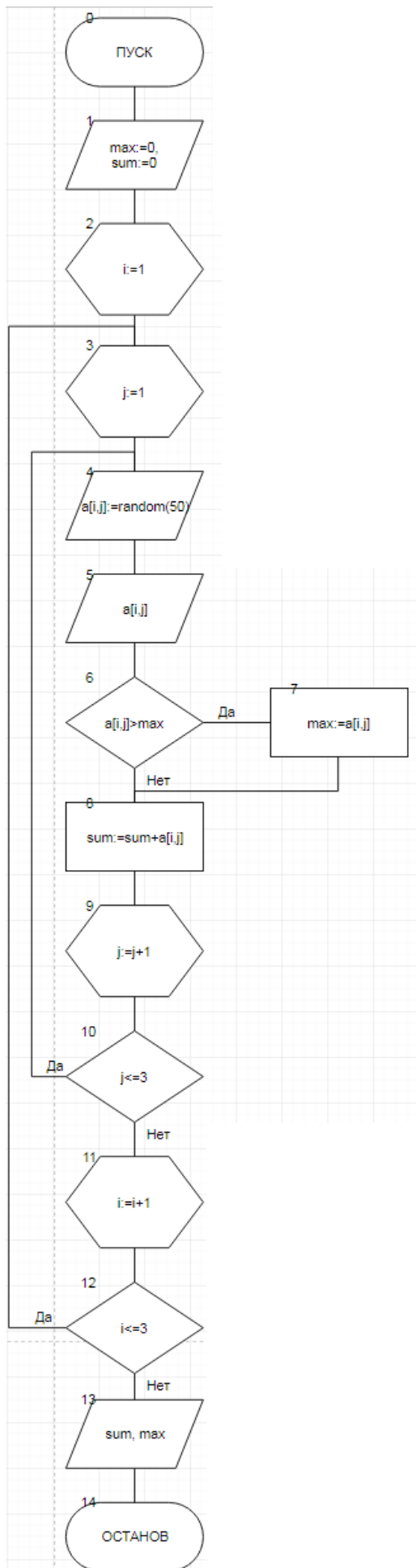
#### **4. Постановка задачи:**

Найти сумму всех элементов массива  $3 \times 3$ . Массив задается явно внутри программы. Найти максимальный элемент.

#### **5. Математическая модель:**

Прибавляем к переменной суммы значение каждого элемента массива. Ищем максимальный.

#### **6. Блок схема:**



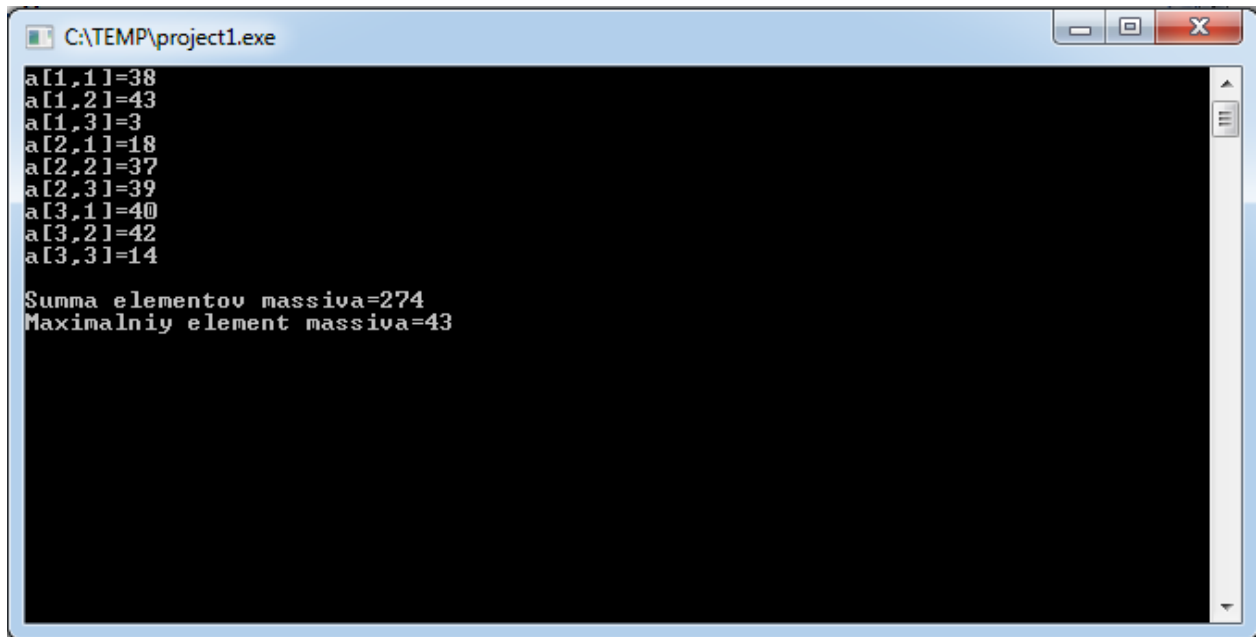
**7. Список идентификаторов:**

| Имя | Смысл                | Тип              |
|-----|----------------------|------------------|
| a   | Числовой массив      | array of integer |
| i   | Параметр цикла       | integer          |
| j   | Параметр цикла       | integer          |
| max | Максимальный элемент | integer          |
| sum | Сумма элементов      | integer          |

**8. Код программы:**

```
program zadanie1;
var
a:array [1..3,1..3] of integer;
i,j,max,sum:integer;
begin
max:=0;
sum:=0;
randomize;
for i:= 1 to 3 do begin
for j:= 1 to 3 do begin
a[i,j]:=random(50);
writeln('a[' ,i ,', ' ,j ,']=',a[i,j]);
if a[i,j]>max then max:=a[i,j];
sum:=sum+a[i,j];
end;
end;
writeln();
writeln('Summa elementov massiva=',sum);
writeln('Maximalniy element massiva=',max);
readln();
end.
```

## 9. Результаты выполненной работы:



```
C:\TEMP\project1.exe
a[1,1]=38
a[1,2]=43
a[1,3]=3
a[2,1]=18
a[2,2]=37
a[2,3]=39
a[3,1]=40
a[3,2]=42
a[3,3]=14
Summa elementov massiva=274
Maximalniy element massiva=43
```

## 10. Анализ результатов вычисления:

В цикле задаем массив, находим максимальный элемент и сумму элементов. Выводим все данные на экран.

### Задача 2

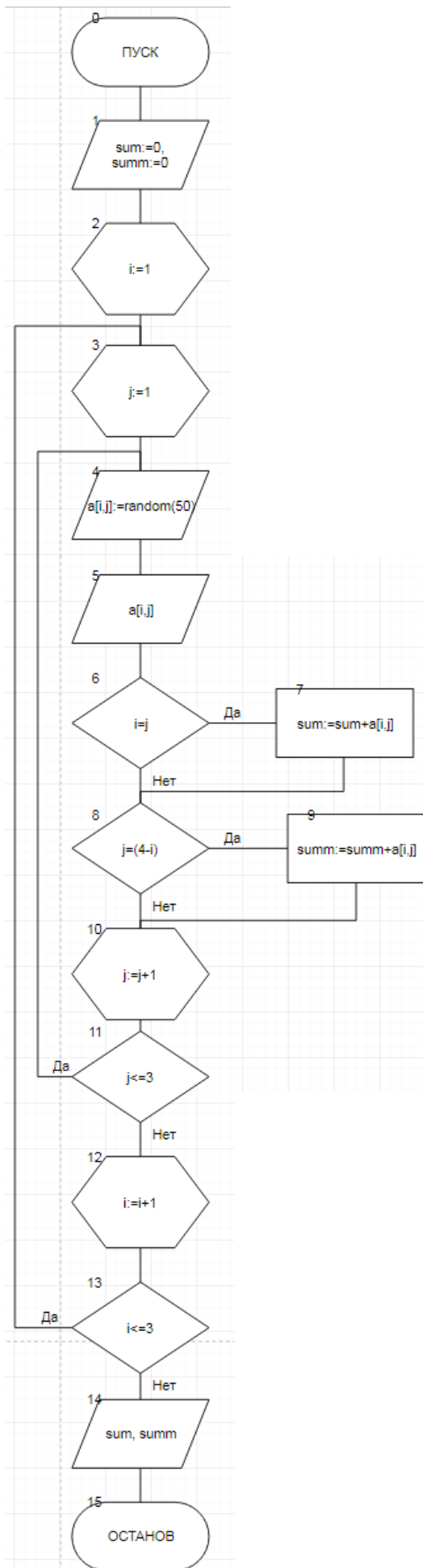
#### 4. Постановка задачи:

Дан массив 3x3. Найти сумму элементов на главной диагонали и сумму элементов побочной диагонали.

#### 5. Математическая модель:

На главной диагонали номер строки равен номеру столбца. На побочной диагонали номер столбца находится по формуле: (размерность массива)-(номер строки)+1.

#### 6. Блок схема:



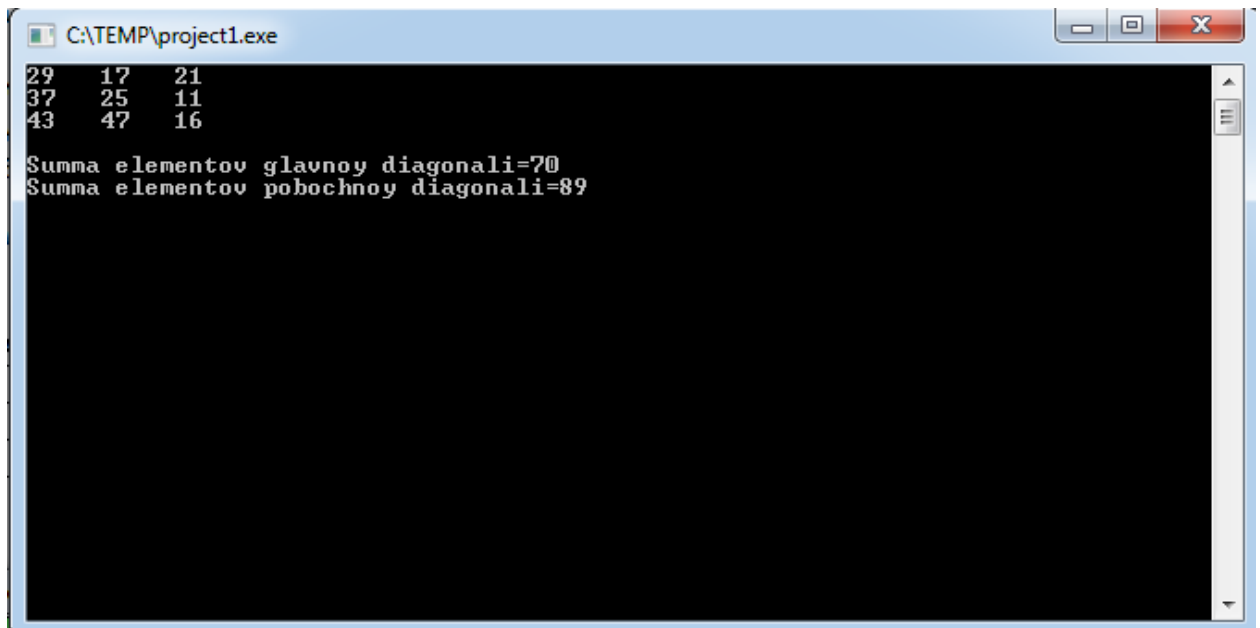
**7. Список идентификаторов:**

| Имя  | Смысл                   | Тип              |
|------|-------------------------|------------------|
| a    | Числовой массив         | array of integer |
| i    | Параметр цикла          | integer          |
| j    | Параметр цикла          | integer          |
| sum  | Сумма на главной диаг.  | integer          |
| summ | Сумма на побочной диаг. | integer          |

**8. Код программы:**

```
program zadanie2;
var
a:array [1..3,1..3] of integer;
i,j,sum,summ:integer;
begin
sum:=0;
summ:=0;
randomize;
for i:=1 to 3 do begin
for j:=1 to 3 do begin
a[i,j]:=random(50);
write(a[i,j], ' ');
if i=j then sum:=sum+a[i,j];
if j=(4-i) then summ:=summ+a[i,j];
end;
writeln();
end;
writeln();
writeln('Summa elementov glavnoy diagonal=' ,sum);
writeln('Summa elementov pobochnoy diagonal=' ,summ);
readln();
end.
```

## 9. Результаты выполненной работы:



```
C:\TEMP\project1.exe
29  17  21
37  25  11
43  47  16

Summa elementov glavnoy diagonali=70
Summa elementov pobochnoy diagonali=89
```

## 10. Анализ результатов вычисления:

Программа задает массив, находит сумму элементов на главной и побочной диагоналях, выводит все данные на экран.

## Задача 3

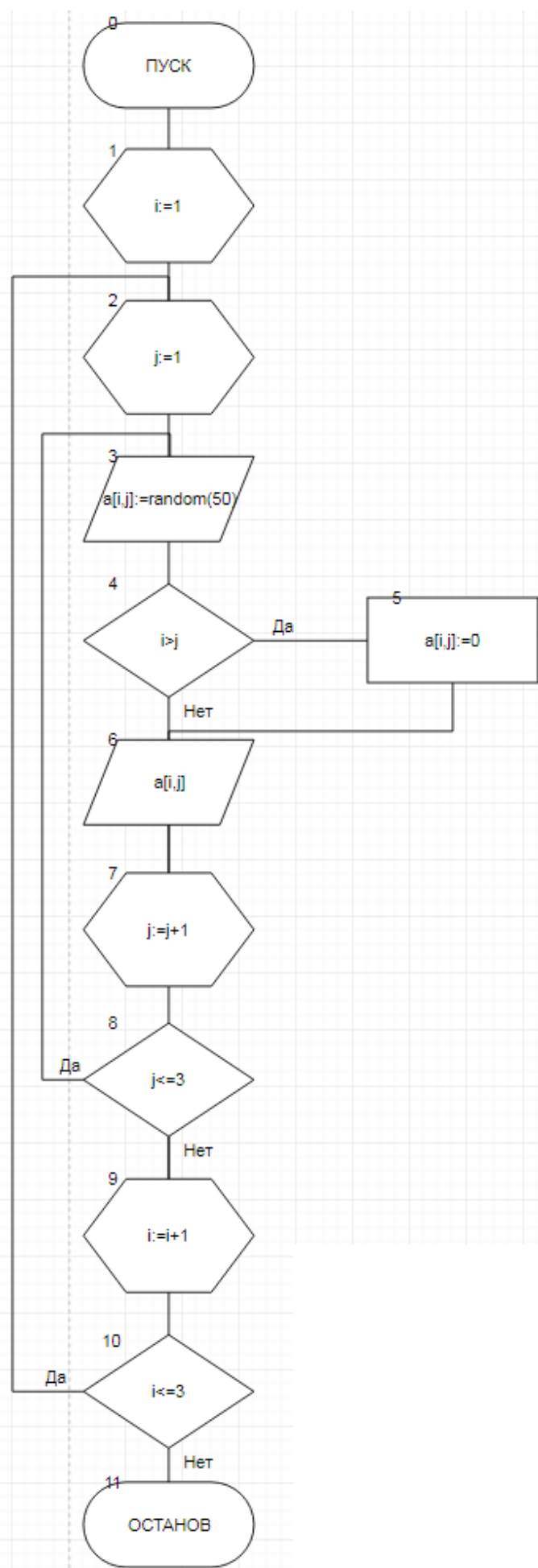
### 4. Постановка задачи:

Дан массив 3x3. Заменить элементы, стоящие ниже главной диагонали нулями.

### 5. Математическая модель:

Если номер строки элемента больше номера столбца, то присваиваем элементу значение 0.

### 6. Блок схема:





### 7. Список идентификаторов:

| Имя | Смысл           | Тип              |
|-----|-----------------|------------------|
| a   | Числовой массив | array of integer |
| i   | Параметр цикла  | integer          |
| j   | Параметр цикла  | integer          |

### 8. Код программы:

```
program zadanie3;
var
a:array [1..3,1..3] of integer;
i,j:integer;
begin
randomize;
for i:=1 to 3 do begin
for j:=1 to 3 do begin
a[i,j]:=random(50);
if i>j then a[i,j] := 0;
write(a[i,j], ' ');
end;
writeln();
end;
readln();
end.
```

### 9. Результаты выполненной работы:



### 10. Анализ результатов вычисления:

Программа задает массив, сразу же заменяет элементы ниже главной диагонали 0, выводит его на экран.

## Задача 4

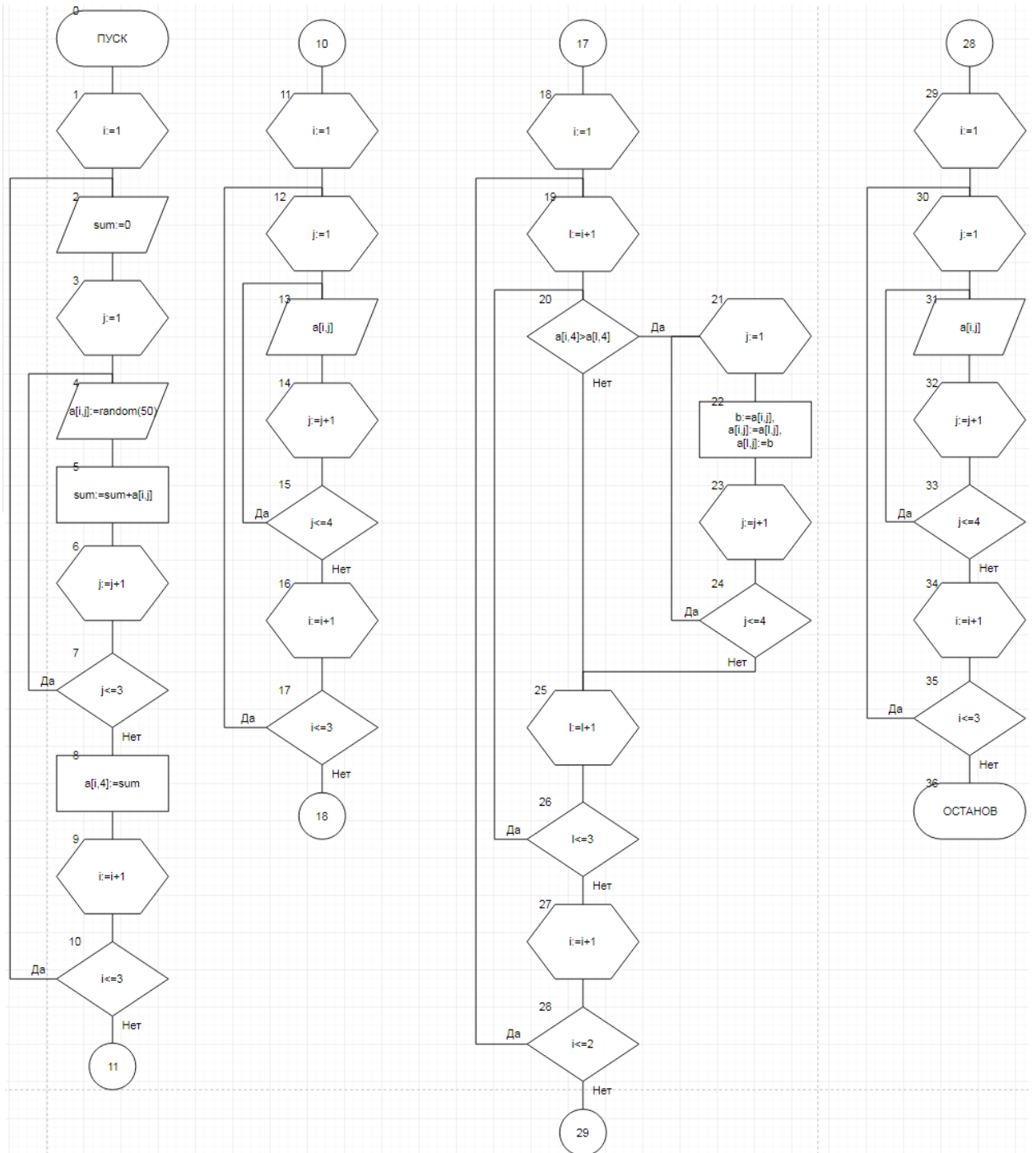
### 4. Постановка задачи:

Дана матрица 3x3. Найти суммы элементов каждой строки и упорядочить строки по возрастанию согласно их суммам.

### 5. Математическая модель:

Находим сумму строк, переставляем их по возрастанию.

### 6. Блок схема:



**7. Список идентификаторов:**

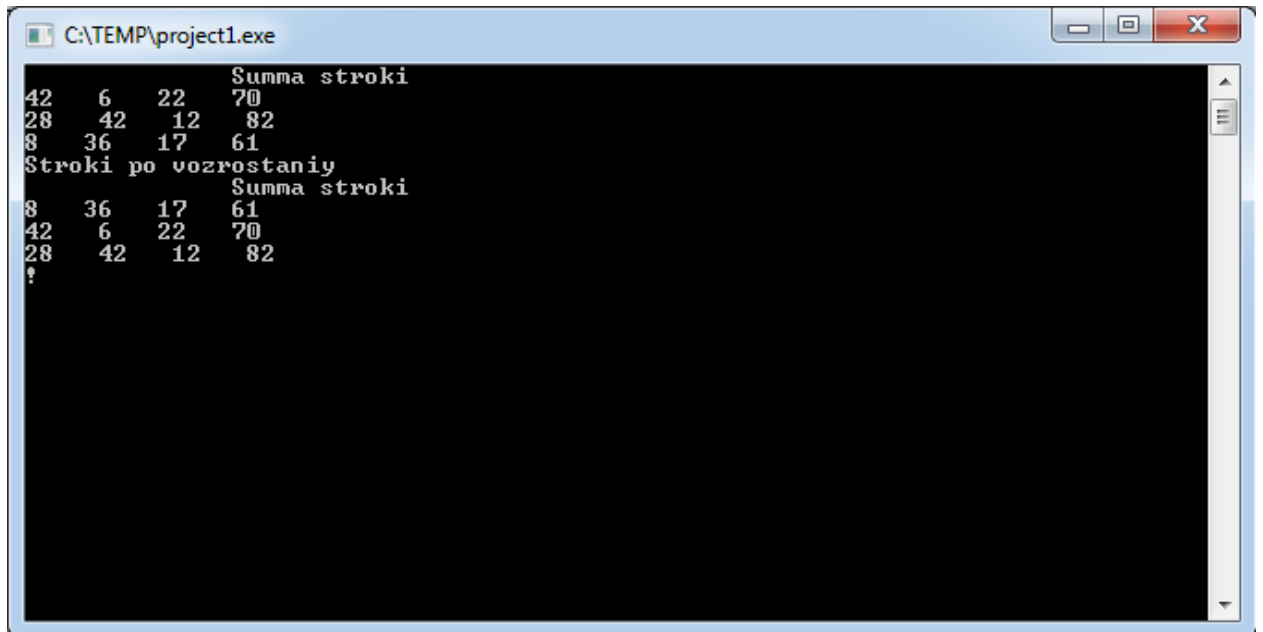
| Имя | Смысл                       | Тип              |
|-----|-----------------------------|------------------|
| a   | Числовой массив             | array of integer |
| i   | Параметр цикла              | integer          |
| j   | Параметр цикла              | integer          |
| l   | Параметр цикла              | integer          |
| sum | Сумма                       | integer          |
| b   | Переменная для перестановки | integer          |

**8. Код программы:**

```
program zadanie4;
var a:array[1..3,1..4] of integer;
i,j,l,sum,b:integer;
begin
  randomize;
  for i:=1 to 3 do
  begin
    sum:=0;
    for j:=1 to 3 do
    begin
      a[i,j]:=random(50);
      sum:=sum+a[i,j];
    end;
    a[i,4]:=sum;
  end;
  writeln('Summa stroki':26);
  for i:=1 to 3 do
  begin
    for j:=1 to 4 do
      write(a[i,j], ' ');
    writeln();
  end;
  for i:=1 to 2 do
    for l:=i+1 to 3 do
      if a[i,4]>a[l,4] then
        for j:=1 to 4 do
          begin
            b:=a[i,j];
            a[i,j]:=a[l,j];
            a[l,j]:=b;
          end;
        writeln('Stroki po vozrastaniy');
        writeln('Summa stroki':26);
```

```
for i:=1 to 3 do
begin
for j:=1 to 4 do
write(a[i,j], ' ');
writeln();
end;
readln();
end.
```

#### 9. Результаты выполненной работы:



```
C:\TEMP\project1.exe
42  6  22  Summa stroki
28  42  12  70
8   36  17  82
Stroki po vozrastaniy
8   36  17  Summa stroki
42  6  22  61
28  42  12  70
?
```

#### 10. Анализ результатов вычисления:

Программа задает массив, где четвертый столбец отведен под сумму строки, выводит его, переставляет в нем строки по возрастанию суммы, выводит массив.

#### 11. Вывод:

Мы изучили комбинированные ВП.