

## Лабораторная работа №12 “Многоступенчатые циклические вычислительные процессы. Двумерные массивы”

### 1 Тема лабораторной работы:

Многоступенчатые циклические вычислительные процессы. Двумерные массивы.

### 2 Цель лабораторной работы:

Реализовать программы для решения поставленных задач с использованием многоступенчатых циклических вычислительных процессов и двумерных массивов средствами PascalABC.net.

### 3 Используемое оборудование:

Персональный компьютер, среда программирования PascalABC.NET

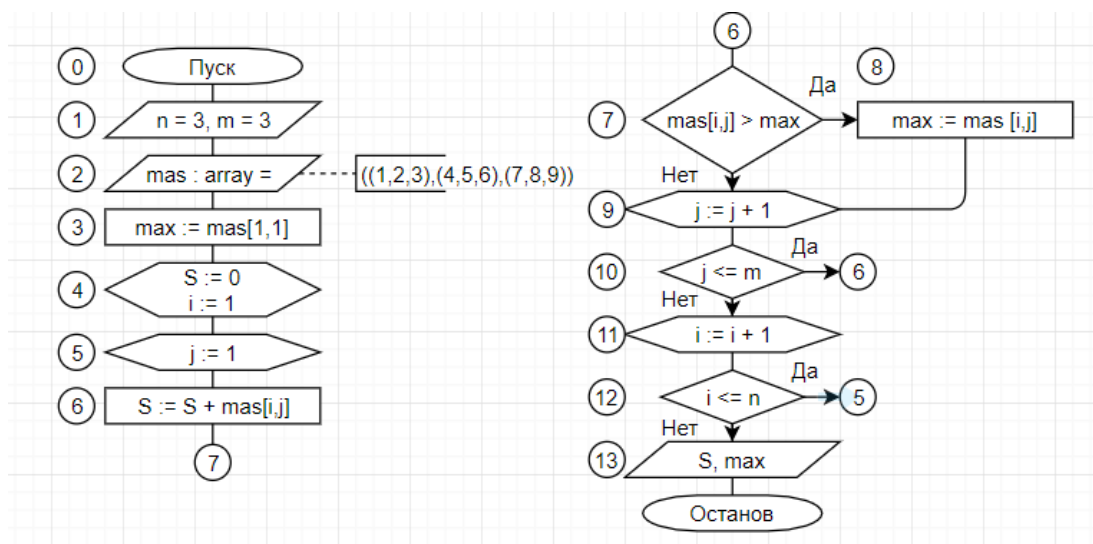
#### 4.1 Постановка задачи

Найти сумму всех элементов массива 3x3. Массив задается явно внутри программы. Найти максимальный элемент.

#### 5.1 Математическая модель

Организуем цикл для перебора всех элементов массива, внутри цикла будем накапливать сумму элементов массива и организуем поиск максимального элемента.

#### 6.1 Блок схема



## 7.1 Список идентификаторов

Имя	Смысл	Тип
n	Количество строк массива	const
m	Количество столбцов массива	const
mas[i,j]	Элемент массива	integer
max	Максимальный элемент массива	integer
S	Сумма элементов массива	integer
i	Номер строки	integer
j	Номер столбца	integer

## 8.1 Код программы

```

program zadacha1;

const n = 3; m = 3;

      mas : array [1..n,1..m] of integer =
_((1,2,3), (4,5,6), (7,8,9));

var i, j, S, max : integer;

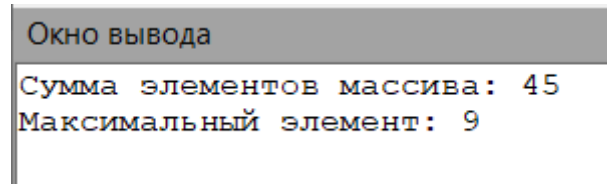
begin
    S := 0;  max := mas[1,1];
    for i:=1 to n do begin
        for j:=1 to m do begin
            S := S + mas [i,j];
            if mas[i,j] > max then max := mas[i,j];
        end;
    end;

    writeln ('Сумма элементов массива: ', S);
    writeln ('Максимальный элемент: ', max);

end.

```

## 9.1 Результат работы программы



## 10.1 Анализ полученных результатов

В ходе решения задачи обработали двумерный массив чисел, подсчитали их сумму и нашли максимальный элемент

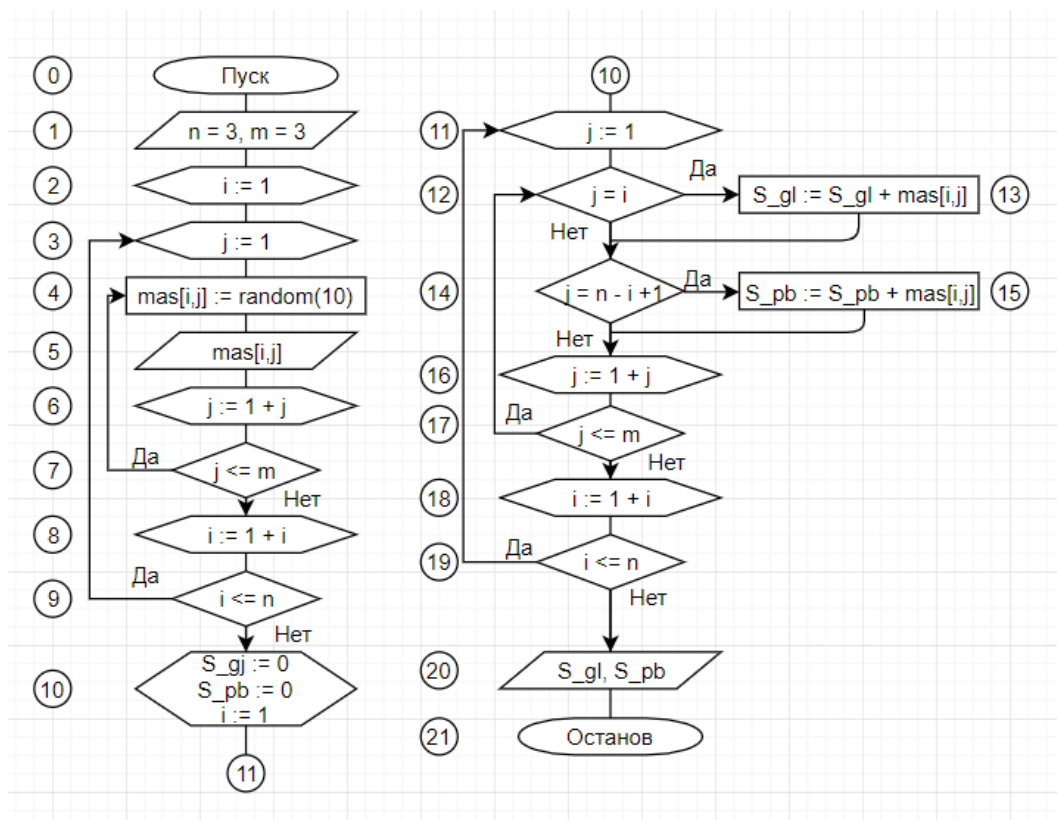
## 4.2 Постановка задачи

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

## 5.2 Математическая модель

В цикле перебираем элементы массивы и при совпадении по условию  $i = j$  будем накапливать сумму элементов главной диагонали, а по условию  $j = n - i + 1$  - элементы побочной диагонали.

## 6.2 Блок схема



## 7.2 Список идентификаторов

Имя	Смысл	Тип
n	Количество строк массива	const
m	Количество столбцов массива	const
mas[i,j]	Элемент массива	integer
S_gl	Сумма элементов главной диагонали	integer
S_pb	Сумма элементов побочной диагонали	integer
i	Номер строки	integer
j	Номер столбца	integer

## 8.2 Код программы

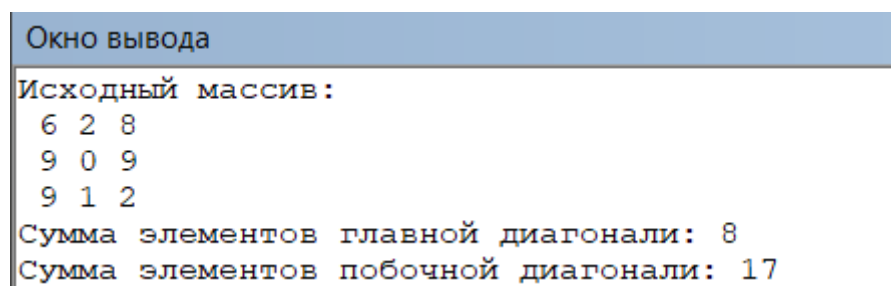
```

program zadacha2;
const n = 3; m = 3;
var i, j, S_gl, S_pb : integer;
    mas : array [1..n,1..m] of integer;
begin
    writeln ('Исходный массив:');
    for i:=1 to n do begin
        for j:=1 to m do begin
            mas[i,j] := random (10);
            write (' ', mas[i,j]);
        end;
        writeln ();
    end;
    S_gl := 0; S_pb := 0;
    for i:=1 to n do begin
        for j:=1 to m do begin
            if (j = i) then S_gl := S_gl + mas[i,j];
            if (j = n - i + 1) then S_pb := S_pb + mas_
_[i,j];
        end;
    end;
end;

```

```
writeln ('Сумма элементов главной диагонали: ', _  
_S_gl);  
writeln ('Сумма элементов побочной диагонали: ', _  
_S_pb);  
end.
```

## 9.2 Результат работы программы



Окно вывода

Исходный массив:

6	2	8
9	0	9
9	1	2

Сумма элементов главной диагонали: 8  
Сумма элементов побочной диагонали: 17

## 10.2 Анализ полученных результатов

Реализовали программу, которая подсчитывает сумму элементов главной и побочной диагонали.

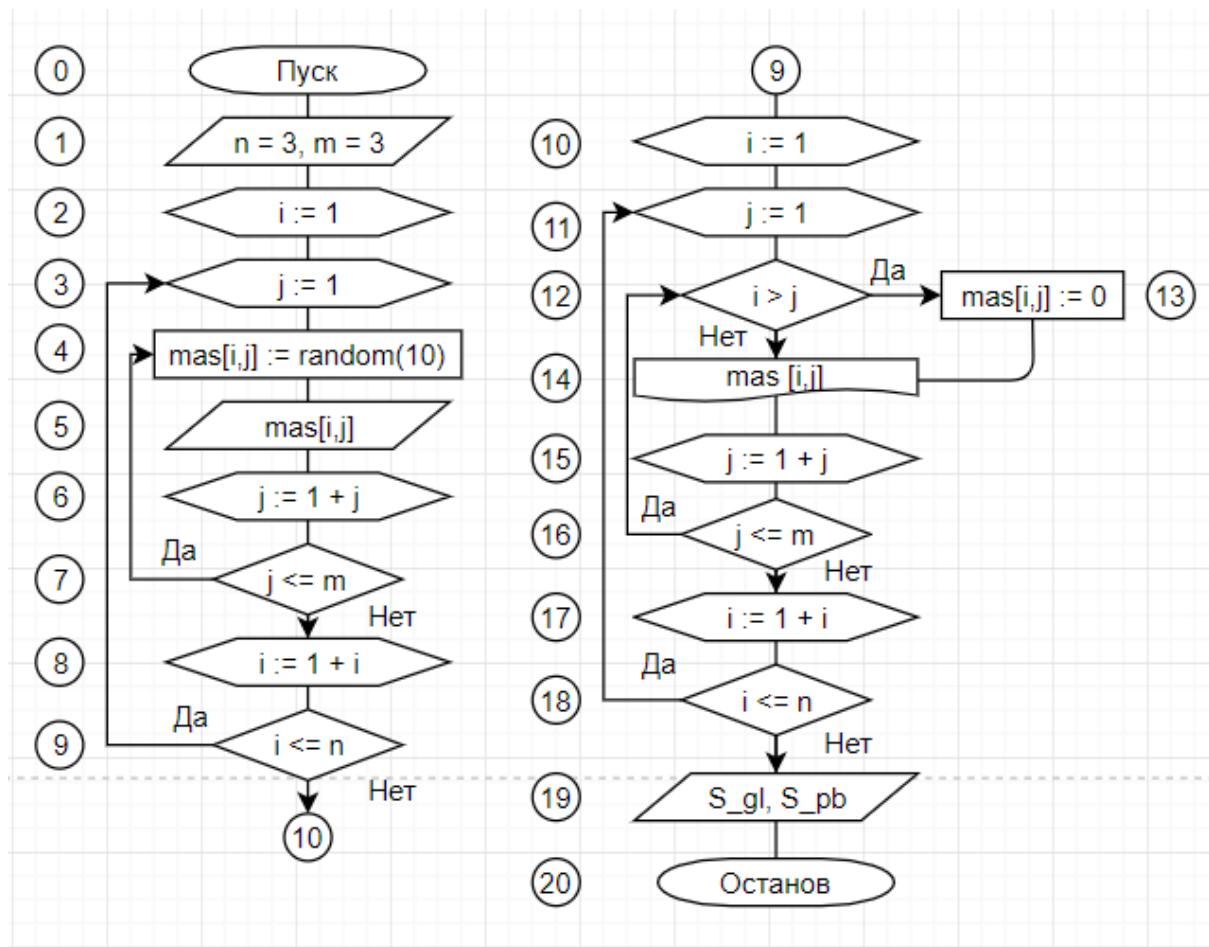
## 4.3 Постановка задачи

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

## 5.3 Математическая модель

У всех элементов стоящих ниже главной диагонали индексы  $i$  всегда больше  $j$ , следуя этому правилу обнулим элементы стоящие ниже главной диагонали.

### 6.3 Блок схема



### 7.3 Список идентификаторов

Имя	Смысл	Тип
n	Количество строк массива	const
m	Количество столбцов массива	const
mas[i,j]	Элемент массива	integer
i	Номер строки	integer
j	Номер столбца	integer

### 8.3 Код программы

```

program zadacha3;
const n = 3; m = 3;
var i, j: integer;

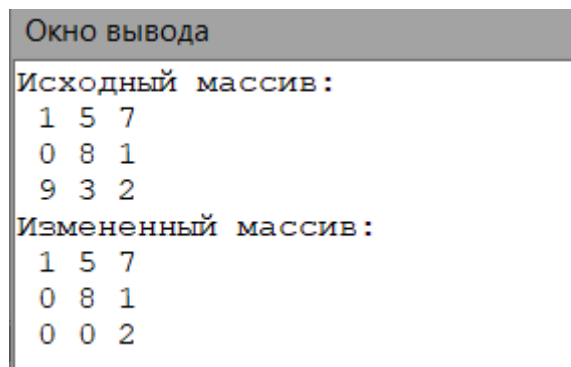
```

```

mas : array [1..n,1..m] of integer;
begin
  writeln ('Исходный массив:');
  for i:=1 to n do begin
    for j:=1 to m do begin
      mas[i,j] := random (10);
      write (' ', mas[i,j]);
    end;
    writeln ();
  end;
  writeln ('Измененный массив:');
  for i:=1 to n do begin
    for j:=1 to m do begin
      if i>j then mas[i,j]:=0;
      write (' ', mas[i,j]);
    end;
    writeln ();
  end;
end.

```

### 9.3 Результат работы программы



```

Окно вывода
Исходный массив:
1 5 7
0 8 1
9 3 2
Измененный массив:
1 5 7
0 8 1
0 0 2

```

### 10.3 Анализ полученных результатов

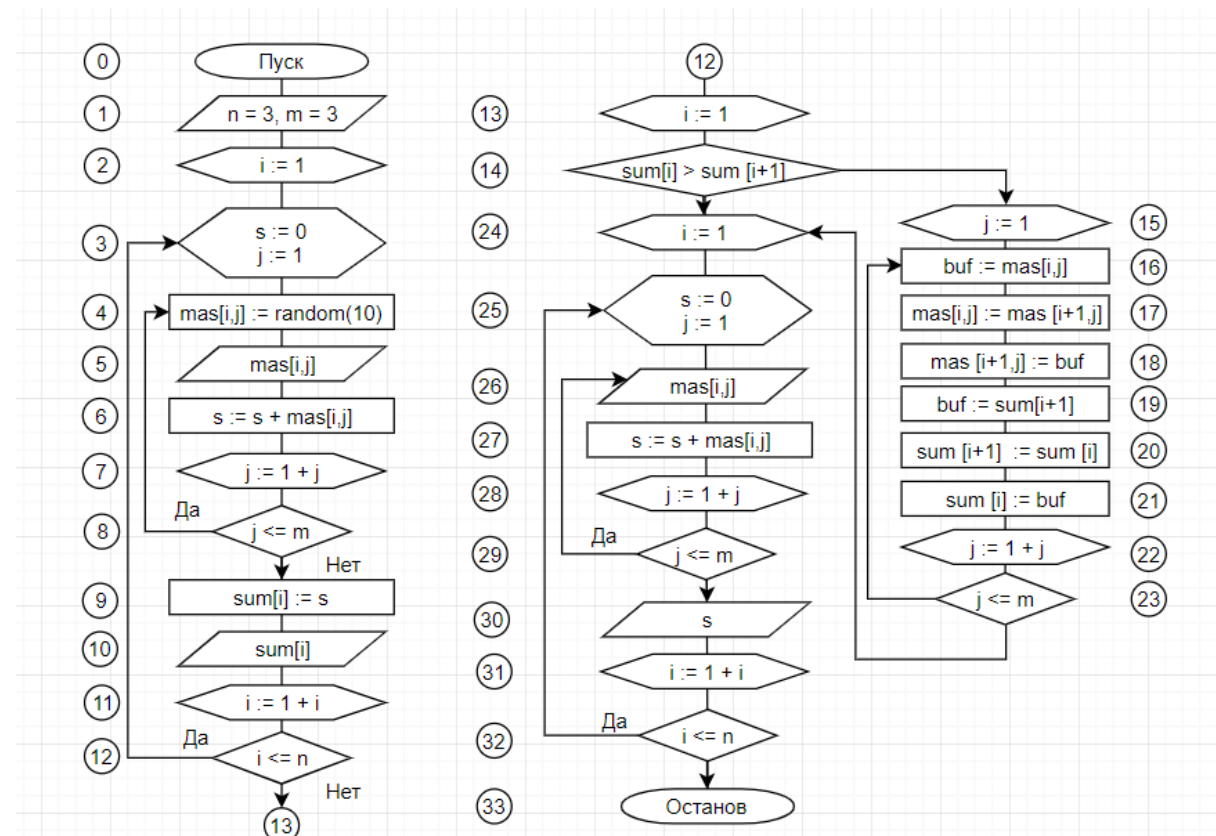
В ходе выполнения задачи перебрали двумерный массив, обнулили все элементы массива стоящие ниже главной диагонали.

#### 4.4 Постановка задачи

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

#### 5.4 Математическая модель

#### 6.4 Блок схема



#### 7.4 Список идентификаторов

Имя	Смысл	Тип
n	Количество строк массива	const
m	Количество столбцов массива	const
mas[i,j]	Элемент массива	integer
i	Номер строки	integer
j	Номер столбца	integer
s	Сумма строки	integer
buf	Буфер	integer



## 8.4 Код программы

```
program zadacha4;
const n = 3; m = 3;
var i, j, s, buf: integer;
mas : array [1..n,1..m] of integer;
sum : array [1..m] of integer;
begin
  writeln ('Исходный массив:');
  for i:=1 to n do begin
    s := 0;
    for j:=1 to m do begin
      mas[i,j] := random (10);
      write (' ', mas[i,j]);
      s := s + mas[i,j]
    end;
    sum[i] := s;
    write (' |', sum[i]);
    writeln ();
  end;

  for i := 1 to n-1 do begin
    if sum[i]>sum[i+1] then
      begin
        for j:=1 to m do begin
          buf := mas[i,j];
          mas [i,j] := mas [i+1,j];
          mas [i+1,j] := buf;
          buf := sum [i+1];
          sum[i+1] := sum [i];
          sum [i] := buf;
        end;
      end;
  end;

  writeln ('Измененный массив:');
```

```
for i:=1 to n do begin
s := 0;
for j:=1 to m do begin
write (' ', mas[i,j]);
s := s + mas[i,j]
end;
write (' |', s);
writeln ();
end;
end.
```

#### 9.4 Результат работы программы



Окно вывода

Исходный массив:

6	5	6	17
1	8	2	11
6	6	3	15

Измененный массив:

1	8	2	11
6	6	3	15
6	5	6	17

#### 10.4 Анализ полученных результатов

При решении данной задачи, реализовали алгоритм сортировки случайного массива по возрастанию суммы строки.

#### 11. Вывод

Изучили методы обработки двумерных массивов. При помощи многоступенчатых ВП реализовали обработку, изменение и сортировку случайных массивов.