Лабораторная работа №12.

**Тема: Многоступенчатые циклические вычислительные процессы.**

**Двумерные массивы.**

**Цель:**

**Используемое оборудование: ПК, среда программирования Lazarus.**

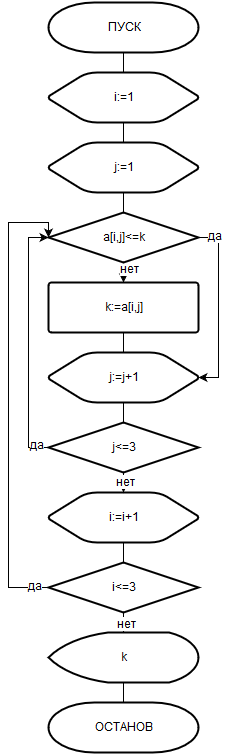
**Задание 1.**

1. Постановка задачи: Найти сумму всех элементов массива 3x3. Массив задается явно внутри программы. Найти максимальный элемент.
2. Математическая модель:

if a[i,j]>k then

k:=a[i,j];

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| j | integer | Параметр цикла |
| k | integer | Максимальный элемент |
| a | matrix | Двумерный массив |

1. **Код программы:**

program zad1;

type

matrix= array [1..3, 1..3] of integer;

var

i,j,k:integer;

a:matrix;

begin

a[1,1]:=1;

a[1,2]:=2;

a[1,3]:=3;

a[2,1]:=4;

a[2,2]:=5;

a[2,3]:=6;

a[3,1]:=7;

a[3,2]:=8;

a[3,3]:=9;

for i:= 1 to 3 do

for j:=1 to 3 do

begin

if a[i,j]>k then

k:=a[i,j];

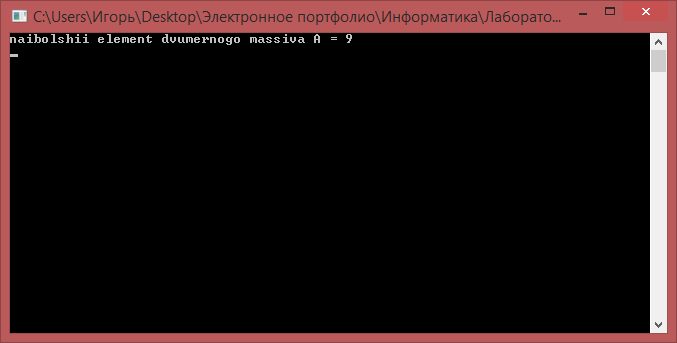
end;

writeln('naibolshii element dvumernogo massiva A = ',k);

readln()

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа выводит максимальный элемент двумерного массива 3х3.
2. Вывод: Программа работает корректно.

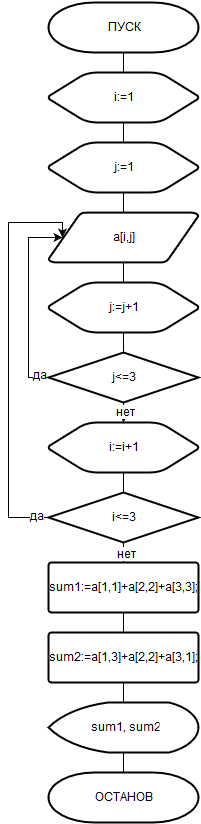
**Задание 2.**

1. Постановка задачи: Дан массив 3x3. Найти сумму элементов на главной диагонали и сумму элементов побочной диагонали.
2. Математическая модель:

sum1:=a[1,1]+a[2,2]+a[3,3];

sum2:=a[1,3]+a[2,2]+a[3,1];

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| j | integer | Параметр цикла |
| a | matrix | Двумерный массив |
| sum1 | integer | Сумма элементов главной диагонали |
| sum2 | integer | Сумма элементов побочной диагонали |

1. **Код программы:**

program zad2;

type

matrix= array [1..3, 1..3] of integer;

var

i,j,sum1,sum2:integer;

a:matrix;

begin

writeln('vvedite elementy massiva A');

for i:=1 to 3 do

for j:=1 to 3 do

readln(a[i,j]);

sum1:=a[1,1]+a[2,2]+a[3,3];

sum2:=a[1,3]+a[2,2]+a[3,1];

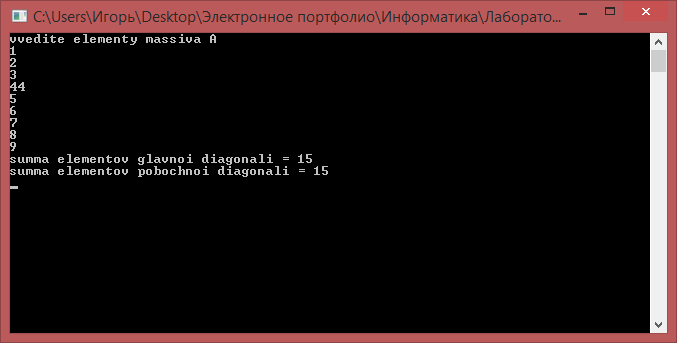
writeln('summa elementov glavnoi diagonali = ',sum1);

writeln('summa elementov pobochnoi diagonali = ',sum2);

readln();

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа вычисляет суммы элементов главной и побочной диагоналей.
2. Вывод: Программа работает корректно.

**Задание 3.**

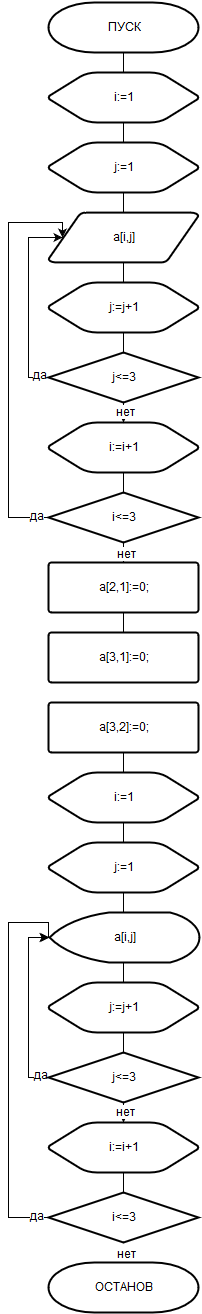
1. Постановка задачи: Дан массив 3x3. Заменить элементы, стоящие ниже главной диагонали нулями.
2. Математическая модель:

a[2,1]:=0;

a[3,1]:=0;

a[3,2]:=0;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| j | Integer | Параметр цикла |
| a | matrix | Двумерный массив |

1. **Код программы:**

program zad3;

type

matrix= array [1..3, 1..3] of integer;

var

i,j:integer;

a:matrix;

begin

writeln('vvedite elementy massiva A');

for i:=1 to 3 do

for j:=1 to 3 do

readln(a[i,j]);

a[2,1]:=0;

a[3,1]:=0;

a[3,2]:=0;

for i:=1 to 3 do

begin

for j:=1 to 3 do

write(a[i,j]:2);

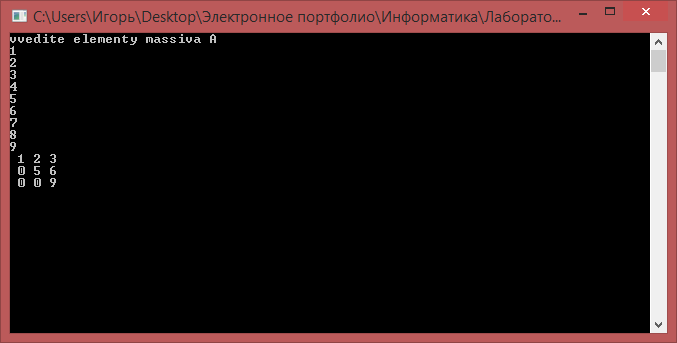
writeln

end;

readln();

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа заменяет элементы стоящие ниже главной диагонали нулями
2. Вывод: Программа работает верно.

**Задание 4.**

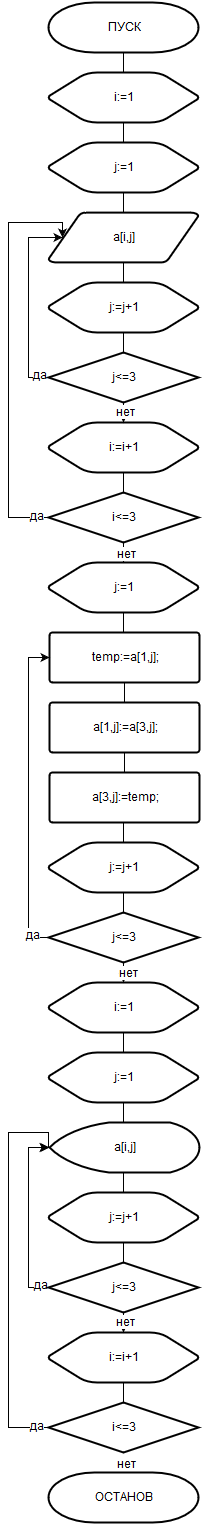
1. Постановка задачи: Дана матрица 3x3. Поменять местами первую и последнюю строки.
2. Математическая модель:

temp:=a[1,j];

a[1,j]:=a[3,j];

a[3,j]:=temp;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | Integer | Параметр цикла |
| j | Integer | Параметр цикла |
| temp | Integer | Временная переменная |
| a | matrix | Двумерный массив |

1. **Код программы:**

program zad4;

type

matrix= array [1..3, 1..3] of integer;

var

i,j,temp:integer;

a:matrix;

begin

writeln('vvedite elementy massiva A');

for i:=1 to 3 do

for j:=1 to 3 do

readln(a[i,j]);

for j:=1 to 3 do

begin

temp:=a[1,j];

a[1,j]:=a[3,j];

a[3,j]:=temp;

end;

for i:=1 to 3 do

begin

for j:=1 to 3 do

write(a[i,j]:2);

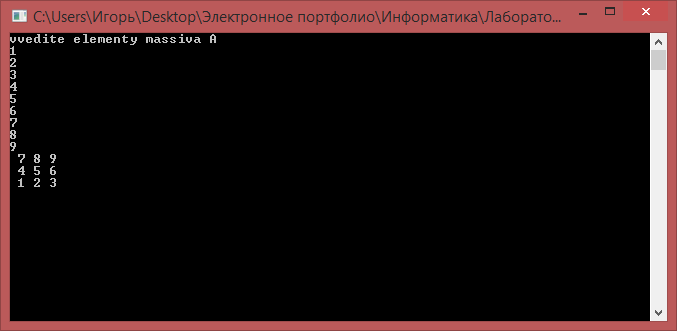
writeln()

end;

readln();

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа меняет местами первую и последнюю строки дввумерного массива 3х3.
2. Вывод: Программа работает корректно.

**Задание 5.**

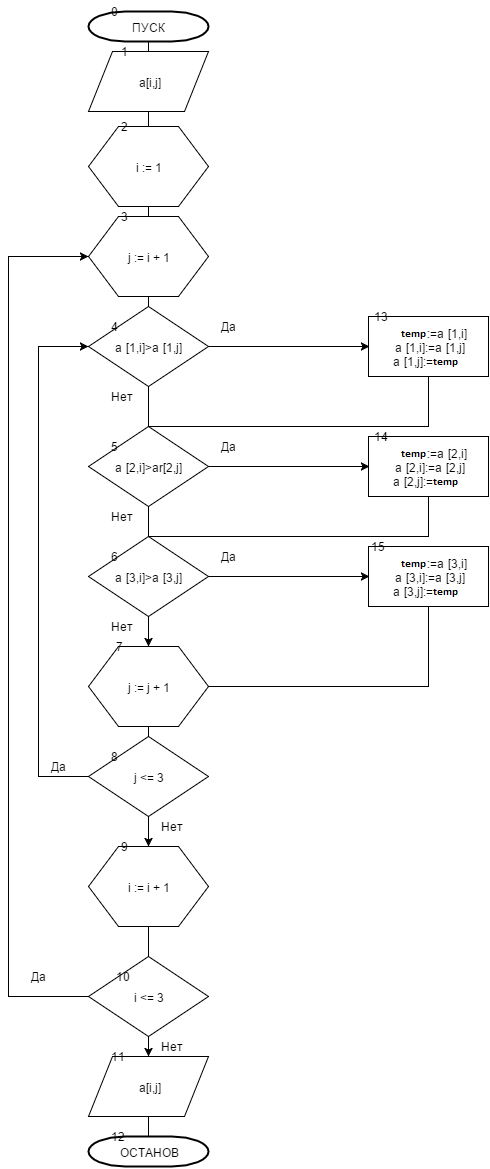
1. Постановка задачи: Дана матрица 3x3. Упорядочить строки по возрастанию.
2. Математическая модель:

temp:=a[1,i];

a[1,i]:=a[1,j];

a[1,j]:=temp;

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | integer | Параметр цикла |
| l | integer | Параметр цикла |
| f | array of integer | Массив |

1. **Код программы:**

Program zad5;

var

a : array [1..3,1..3] of integer;

i,j,temp: integer;

begin

randomize;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

a[i,j] := random(50);

write(a[i,j],' ');

end;

writeln;

end;

for i:=1 to 3 do

for j:=i+1 to 3 do begin

if a[1,i]>a[1,j] then

begin

temp:=a[1,i];

a[1,i]:=a[1,j];

a[1,j]:=temp;

end;

if a[2,i]>a[2,j] then

begin

temp:=a[2,i];

a[2,i]:=a[2,j];

a[2,j]:=temp;

end;

if a[3,i]>a[3,j] then

begin

temp:=a[3,i];

a[3,i]:=a[3,j];

a[3,j]:=temp;

end;

end;

writeln('Otsortirovannyi massiv: ');

for i := 1 to 3 do begin

for j := 1 to 3 do

write(a[i,j],' ');

writeln;

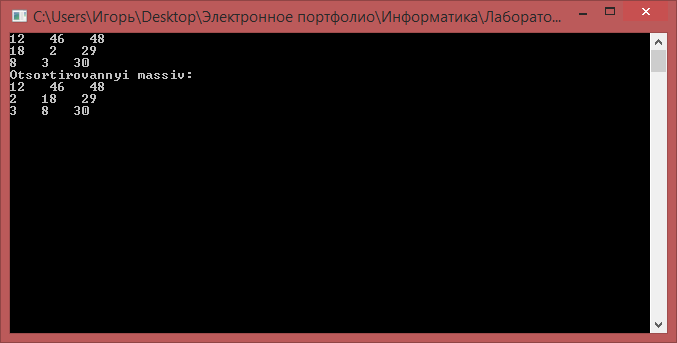
end;

readln();

readln;

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программа сортирует каждую строку двумерного массива по возрастанию.
2. Вывод: программа работает корректно

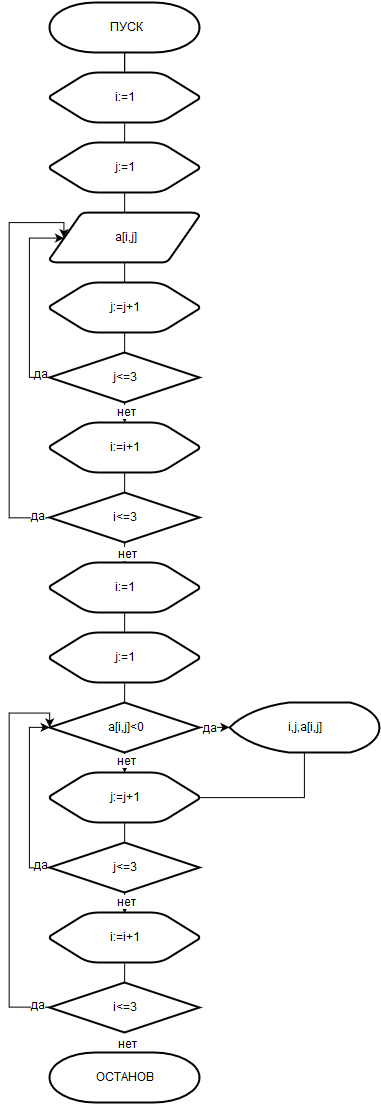
**Задание 6.**

1. Постановка задачи: Дана матрица 3x3. Выдать на экран все элементы, значение которых меньше 0, номер этого элемента в матрице
2. Математическая модель:

if a[i,j]<0 then

writeln('a[',i,',',j,'] = ',a[i,j]);

1. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| i | Integer | Параметр цикла |
| j | Integer | Параметр цикла |
| a | matrix | Двумерный массив |

1. **Код программы:**

program zad6;

type

matrix= array [1..3, 1..3] of integer;

var

i,j:integer;

a:matrix;

begin

writeln('vvedite elementy massiva A');

for i:=1 to 3 do

for j:=1 to 3 do

readln(a[i,j]);

for i:=1 to 3 do

for j:=1 to 3 do

begin

if a[i,j]<0 then

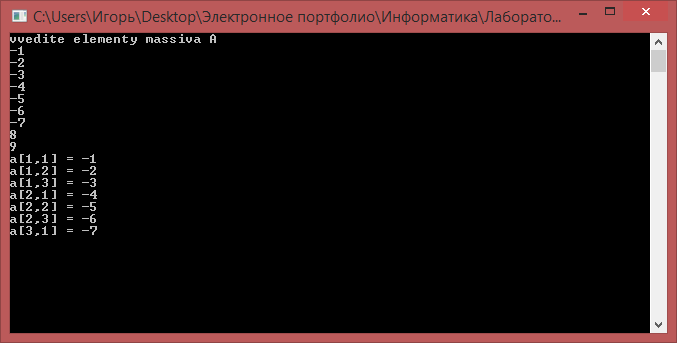
writeln('a[',i,',',j,'] = ',a[i,j]);

end;

readln()

end.

1. Результат выполнения программы:



1. Анализ результатов вычисления: Программма выдаёт на экран все элементы, значение которых меньше 0 и позицию этого элемента в матрице.
2. Вывод: программа работает корректно