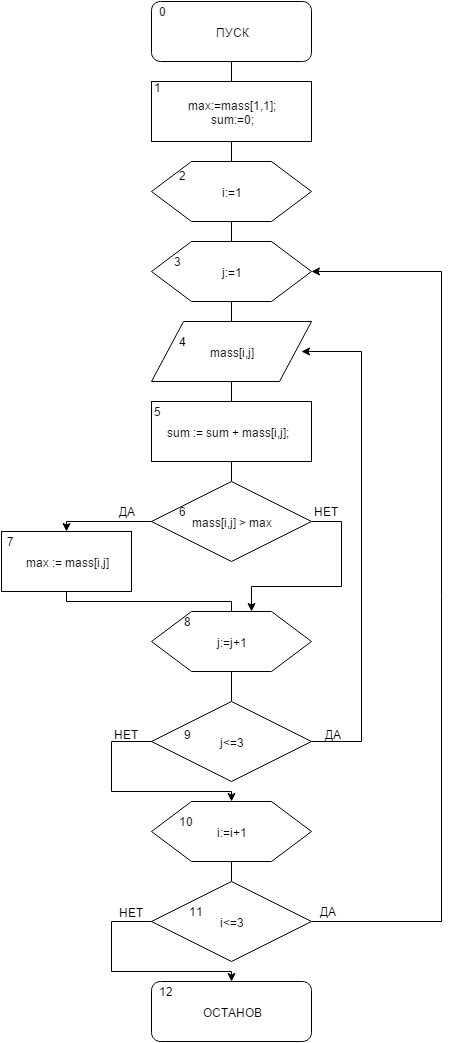
Лабораторная работа №12.

*Задание 1.*

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: найти сумму всех элементов массива 3x3. Массив задается явно внутри программы. Найти максимальный элемент.
5. Математическая модель: будем прибавлять к переменной суммы значение каждого элемента массива. Перебирая все элементы массива, ищем максимальный.
6. Блок-схема: ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |
| max | integer | Максимальный элемент массива |
| sum | integer | Сумма элементов |

1. Код программы:

program zad1;

var

mass: array [1..3,1..3] of longint=((2,5,8),(6,7,3),(1,2,9));

var

i,j, max, sum :longint;

begin

max:=mass[1,1];

sum:=0;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

sum := sum + mass[i,j];

if mass[i,j] > max then max := mass[i,j];

end;

writeln();

end;

writeln();

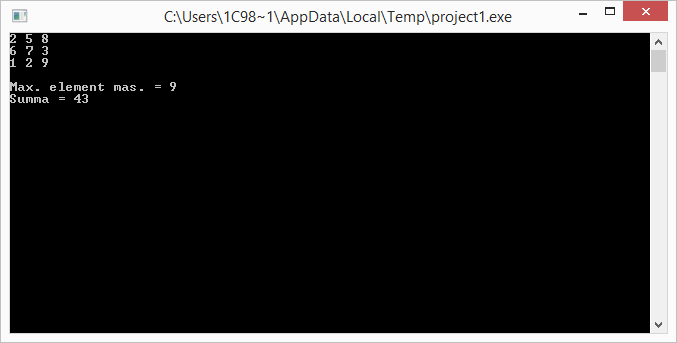
writeln('Max. element mas. = ',max);

writeln('Summa = ',sum);

readln;

end.

1. Результат работы программы:



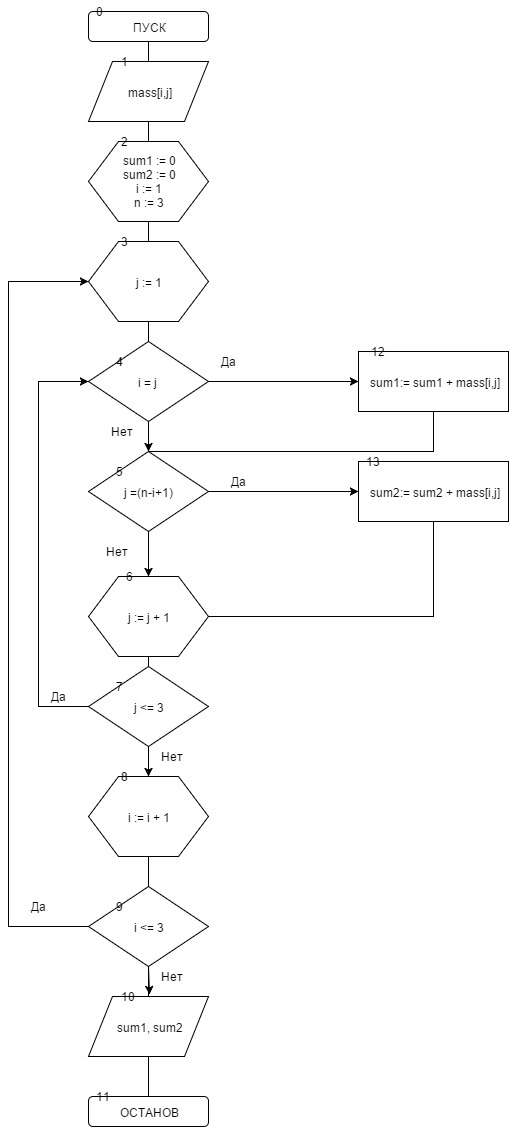
1. Анализ результатов вычисления:

В цикле задаем массив, здесь же прибавляем к переменной суммы значение очередного элемента. Затем, перебирая каждый элемент массива, ищем максимальный.

1. Вывод: программа работает корректно.

*Задание 2.*

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
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 номер столбца можно найти так: n-i+1, где n – размерность массива, I – номер строки.
6. Блок-схема: ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |
| sum1 | integer | Сумма элементов на главной диагонали |
| Sum2 | integer | Сумма элементов на побочной диагонали |
| n | integer | Размер массива |

1. Код программы:

program task2;

var

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

i,j, sum2, sum1,n : integer;

begin

randomize;

n := 3;

for i := 1 to n do begin

for j := 1 to n do begin

mass[i,j] := random(100);

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

end;

writeln;

end;

for i := 1 to n do begin

for j := 1 to n do begin

if i = j then sum1 := sum1+ mass[i,j];

if j = (n-i+1) then sum2 := sum2 + mass[i,j];

end;

end;

writeln();

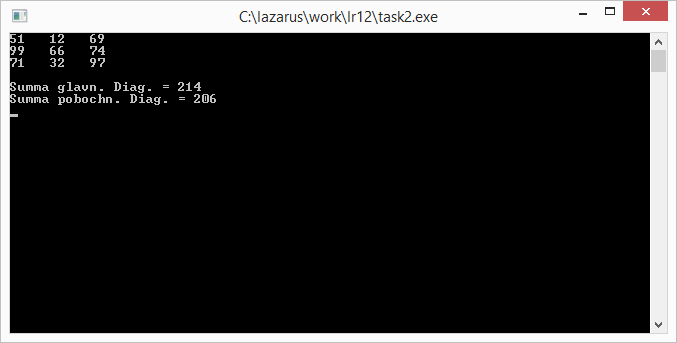
writeln('Summa glavn. Diag. = ',sum1);

writeln('Summa pobochn. Diag. = ',sum2);

readln;

end.

1. Результат работы программы:



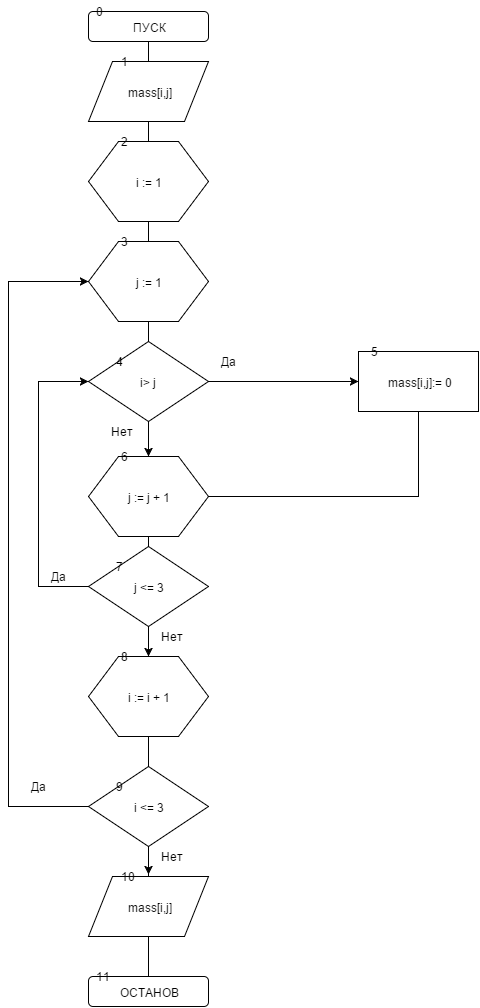
1. Анализ результатов вычисления:

Суммируем сначала все элементы, стоящие на главной диагонали, затем все на побочной.

1. Вывод: программа работает корректно.

*Задание 3.*

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дан массив 3x3. Заменить элементы, стоящие ниже главной диагонали нулями.
5. Математическая модель: если номер строки больше номера столбца, то присваиваем элементу массива значение 0.
6. Блок-схема: ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |

1. Код программы:

program task3;

var

mass : 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

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

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

end;

writeln;

end;

writeln();

for i := 1 to 3 do

for j := 1 to 3 do

if i > j then mass[i,j] := 0;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

end;

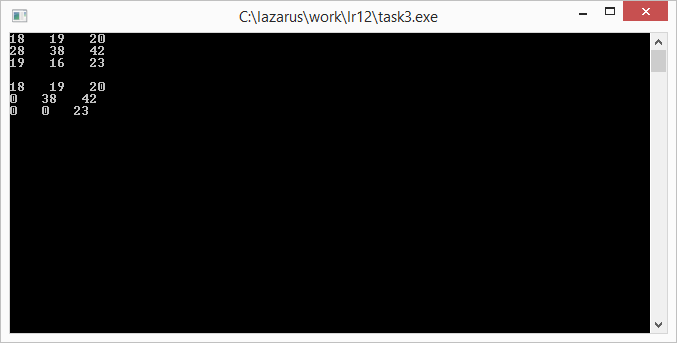
writeln;

end;

readln;

end.

1. Результат работы программы:



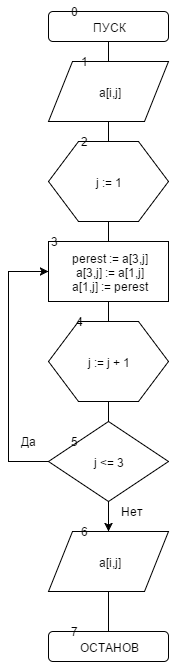
1. Анализ результатов вычисления:

Перебирая каждый элемент, проверяем следующее условие: больше ли номер строки номера столбца. Если да, то присваиваем элементу массива значение 0.

1. Вывод: программа работает корректно.

*Задание 4.*

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дана матрица 3x3. Поменять местами первую и последнюю строки.
5. Математическая модель: берем сначала «все столбы» первой строки и меняем местами со «всеми столбцами» третьей строки.
6. Блок-схема: ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |
| per | integer | Переменная для перестановки |

1. Код программы:

program task4;

var

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

i,j, per: integer;

begin

randomize;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

mass[i,j] := random(70);

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

end;

writeln;

end;

writeln();

for j := 1 to 3 do begin

per := mass[3,j];

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

mass[1,j] := per;

end;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

end;

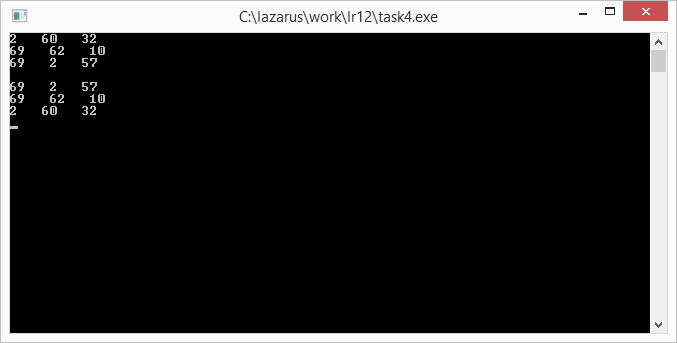
writeln;

end;

readln;

end.

1. Результат работы программы:



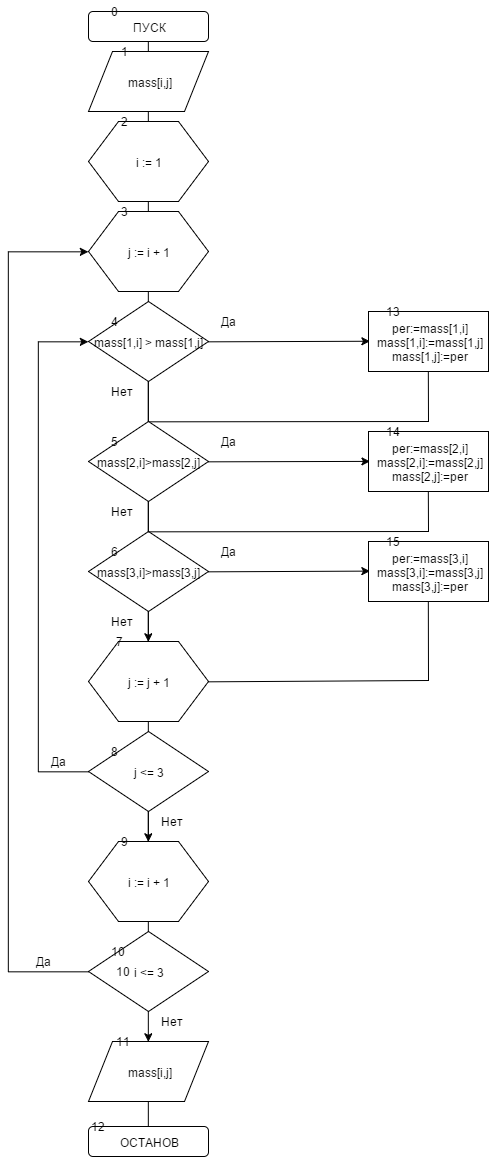
1. Анализ результатов вычисления:

Меняем посредством перестановки столбцов в нужных строках.

1. Вывод: программа работает корректно.

Задание 5.

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дана матрица 3x3. Упорядочить строки по возрастанию.
5. Математическая модель:
6. Блок-схема: ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |
| per | integer | Переменная для перестановки |

1. Код программы:

program task5;

var

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

i,j,per: integer;

begin

randomize;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

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

end;

writeln;

end;

writeln;

writeln;

for i:=1 to 3 do

for j:=i+1 to 3 do begin

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

begin

per:= mass[1,i];

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

mass[1,j]:= per;

end;

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

begin

per:= mass[2,i];

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

mass[2,j]:= per;

end;

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

begin

per:= mass[3,i];

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

mass[3,j]:= per;

end;

end;

for i := 1 to 3 do begin

for j := 1 to 3 do

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

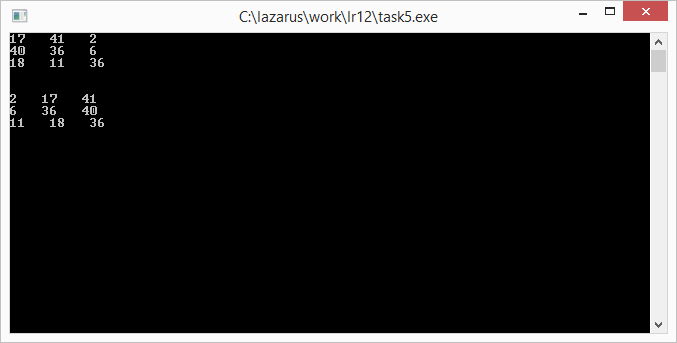
writeln;

end;

readln();

end.

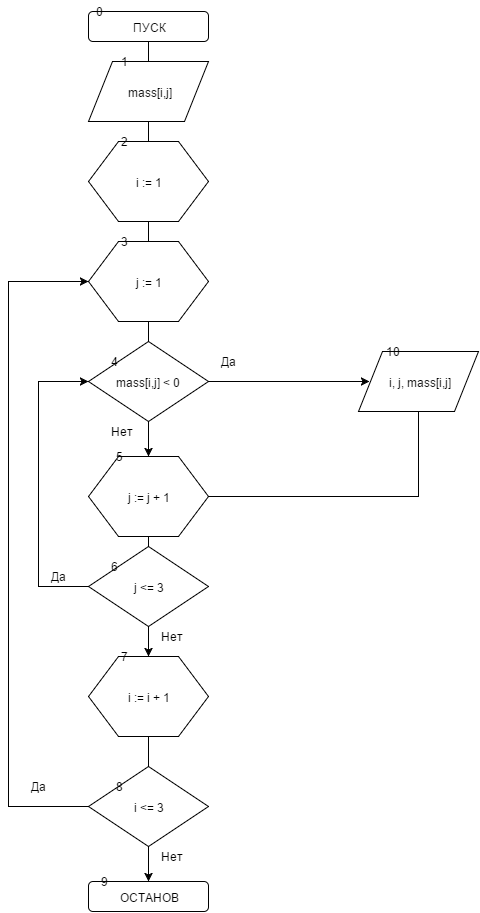
1. Результат работы программы:



1. Анализ результатов вычисления: используем сортировку методом выбора для каждой строки.
2. Вывод: программа работает корректно.

Задание 6.

1. Тема: Многоступенчатые циклические вычислительные процессы. Двумерные массивы.
2. Цель: научиться реализовывать многоступенчатые циклические вычислительные процессы средствами FreePascal.
3. Используемое оборудование: ПК, среда программирования Lazarus.
4. Постановка задачи: дана матрица 3x3. Выдать на экран все элементы, значение которых меньше 0, номер этого элемента в матрице.
5. Математическая модель: проверяем поэлементно – если значение элемента < 0, то выводим сам элемент и его индекс.
6. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| mass | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |

1. Код программы:

program task6;

var

mass : 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

mass[i,j] := random(30)-15;

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

end;

writeln;

end;

writeln;

for i := 1 to 3 do

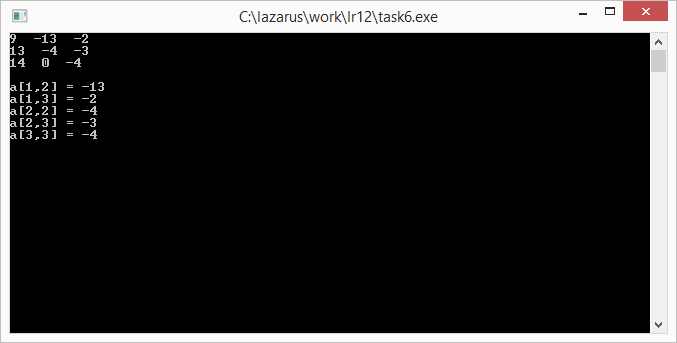
for j := 1 to 3 do

if mass[i,j] < 0 then writeln('a[',i,',',j,'] = ',mass[i,j]);

readln;

end.

1. Результат работы программы:



1. Анализ результатов вычисления: проверяем – если значение элемента меньше 0, то выводим на экран элемент и его индекс.
2. Вывод: программа работает корректно.