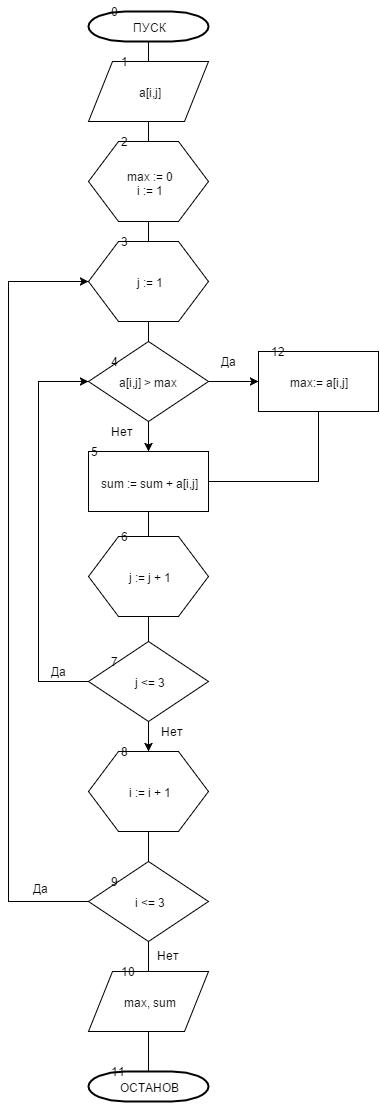
Лабораторная работа №12.

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: см.ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | 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 lr12\_task1;

var

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

i,j, max, sum : integer;

begin

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]);

end;

end;

max := 0;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

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

end;

end;

writeln();

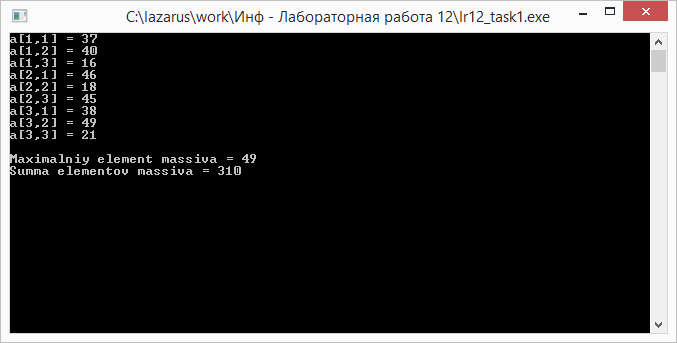
writeln('Maximalniy element massiva = ',max);

writeln('Summa elementov massiva = ',sum);

readln;

end.

1. Результат работы программы:



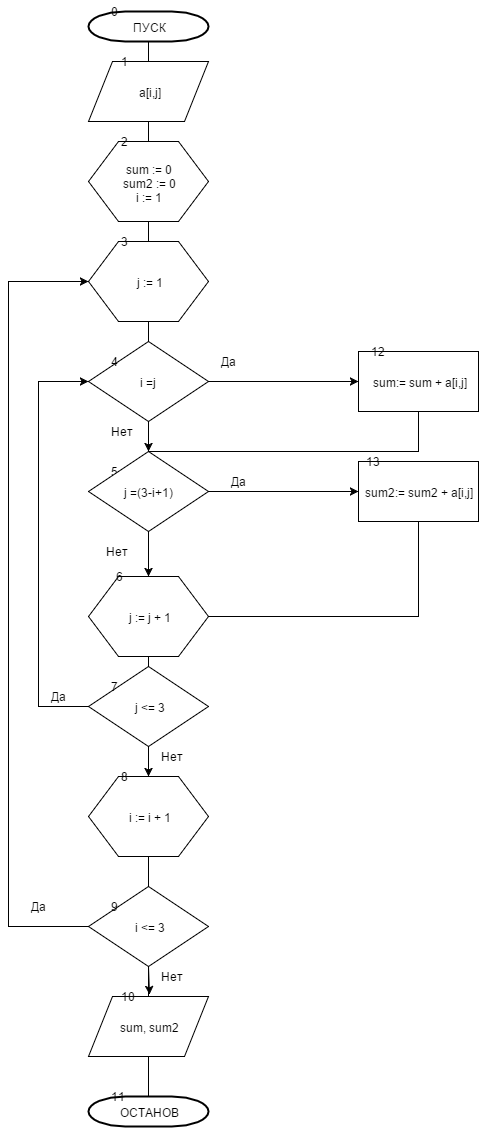
1. Анализ результатов вычисления:

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: см. ниже



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Смысл |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |
| sum | integer | Сумма элементов на главной диагонали |
| Sum2 | integer | Сумма элементов на побочной диагонали |

1. Код программы:

program lr12\_task2;

var

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

i,j, sum2, sum : integer;

begin

sum := 0;

sum2 := 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],' ');

end;

writeln;

end;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

if i = j then sum := sum + a[i,j];

if j = (3-i+1) then sum2 := sum2 + a[i,j];

end;

end;

writeln();

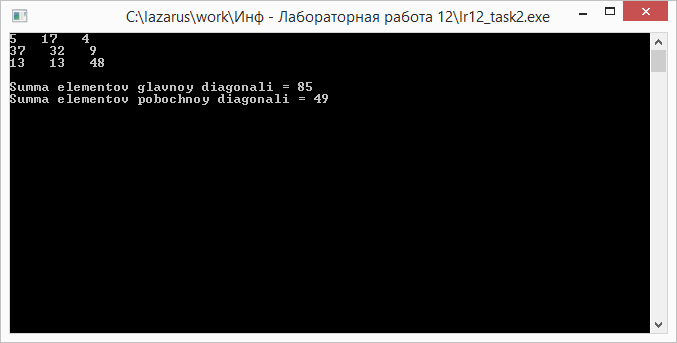
writeln('Summa elementov glavnoy diagonali = ',sum);

writeln('Summa elementov pobochnoy diagonali = ',sum2);

readln;

end.

1. Результат работы программы:



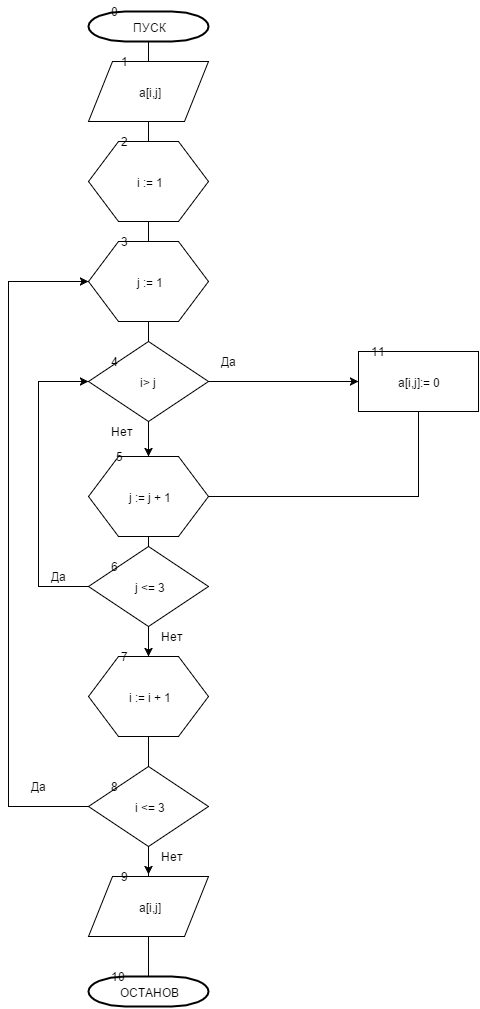
1. Анализ результатов вычисления:

Суммируем сначала все элементы 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:



1. Список идентификаторов:

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

1. Код программы:

program lr12\_task3;

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);

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

end;

writeln;

end;

writeln();

for i := 1 to 3 do

for j := 1 to 3 do

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

writeln('Zamenim elementy nije glavnoy diagonali 0');

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

end;

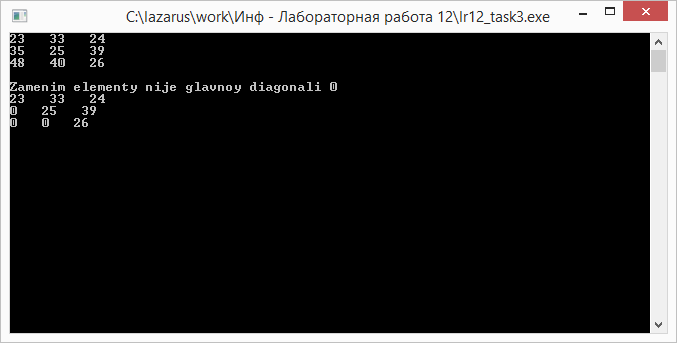
writeln;

end;

readln;

end.

1. Результат работы программы:



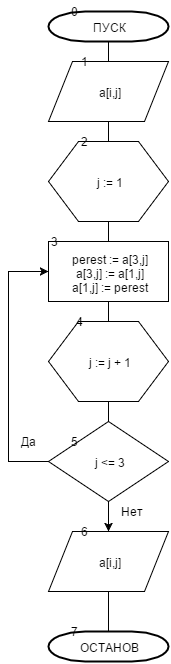
1. Анализ результатов вычисления:

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:



1. Список идентификаторов:

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

1. Код программы:

program lr12\_task4;

var

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

i,j,perest : 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;

writeln();

for j := 1 to 3 do begin

perest := a[3,j];

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

a[1,j] := perest;

end;

writeln('Pomenyem mestami pervuy i poslednyy stroki');

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

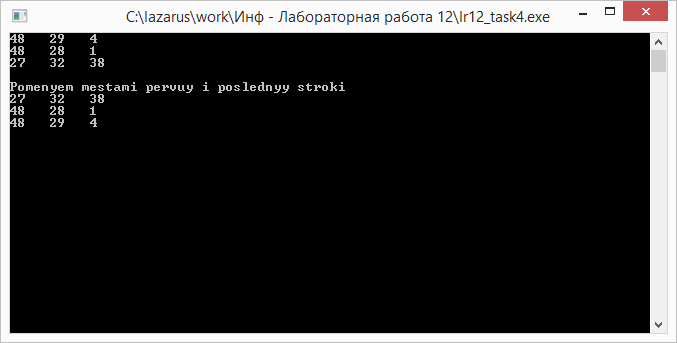
end;

writeln;

end;

readln;

end.

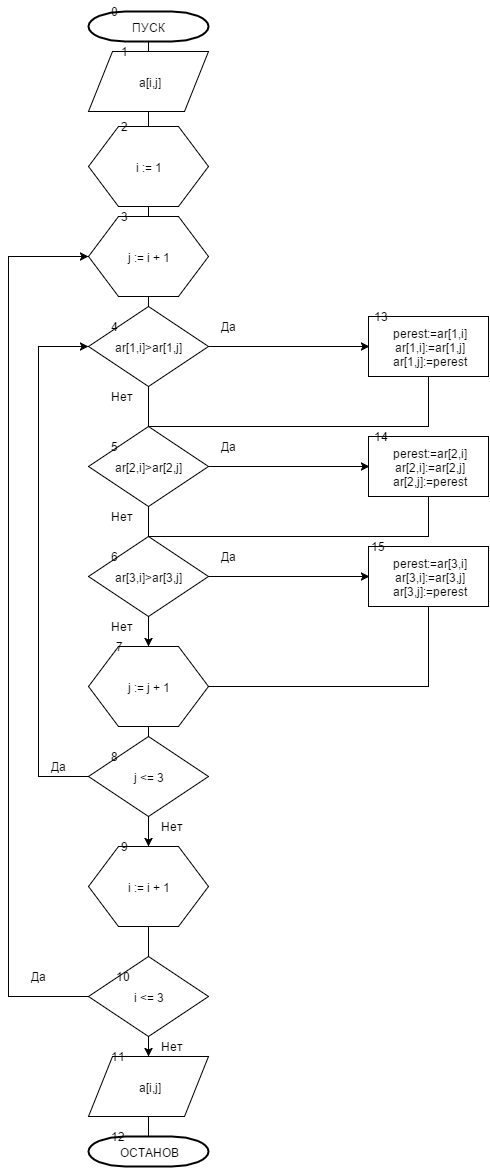
1. Результат работы программы: 
2. Анализ результатов вычисления:

берем сначала «все столбы» первой строки и меняем местами со «всеми столбцами» третьей строки.

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:



1. Список идентификаторов:

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

1. Код программы:

program lr12\_task5;

var

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

i,j,perest: integer;

begin

randomize;

for i := 1 to 3 do begin

for j := 1 to 3 do begin

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

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

end;

writeln;

end;

writeln;

writeln('Sortiruem po vozrastaniy: ');

writeln();

for i:=1 to 3 do

for j:=i+1 to 3 do begin

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

begin

perest:=ar[1,i];

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

ar[1,j]:=perest;

end;

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

begin

perest:=ar[2,i];

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

ar[2,j]:=perest;

end;

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

begin

perest:=ar[3,i];

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

ar[3,j]:=perest;

end;

end;

for i := 1 to 3 do begin

for j := 1 to 3 do

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

writeln;

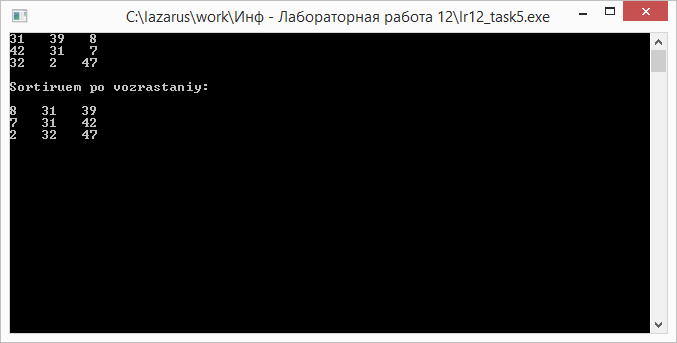
end;

readln();

readln;

end.

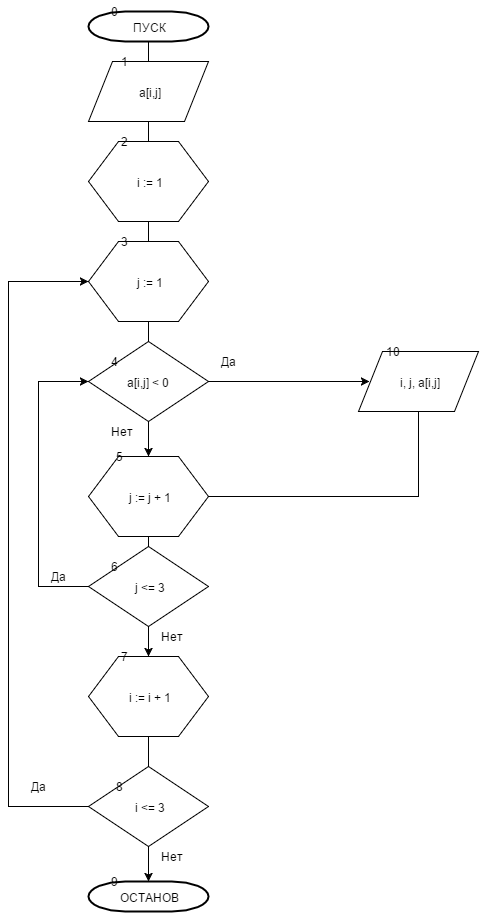
1. Результат работы программы:



1. Анализ результатов вычисления: используем сортировку методом выбора для каждой строки отдельно.
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 |
| a | Array of integer | Числовой массив |
| i | integer | Параметр цикла, индекс |
| j | integer | Параметр цикла, индекс |

1. Код программы:

program lr12\_task6;

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)-25;

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

end;

writeln;

end;

writeln;

for i := 1 to 3 do

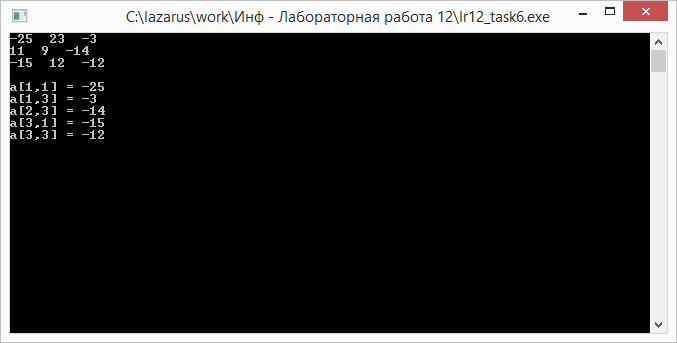
for j := 1 to 3 do

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

readln;

end.

1. Результат работы программы:



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