**Лабораторная работа №12**

**Многоступенчатые циклические вычислительные**

**процессы. Двумерные массивы.**

**Цель:** научиться реализовывать алгоритмы многоступенчатых циклических вычислительных процессов средствами компилятора free Pascal.

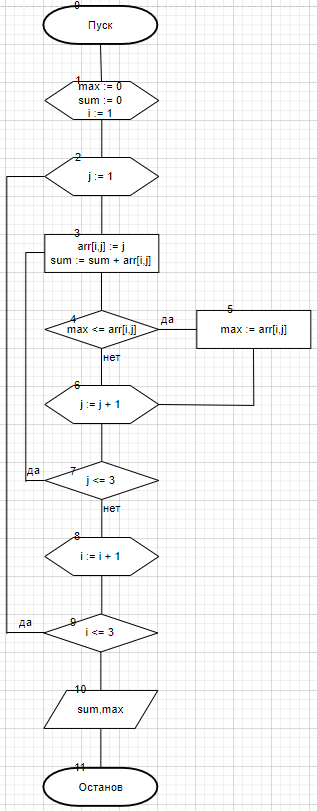
**Используемое оборудование:** пк, PascalABC.

**Задача 1:**

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

**5.1)** sum := sum + arr[i,j], i := i + 1, j := j + 1

**6.1)**



**7.1)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **i** | **Параметр внешнего цикла** | **int** |
| **j** | **Параметр внутреннего цикла** | **int** |
| **max** | **Максимальный элемент массива arr** | **int** |
| **sum** | **Накопительная переменная, сумма всех элементов масиива** | **int** |
| **arr[3,3]** | **Двумерный массив размерностью 3х3** | **int** |

**8.1)**

**program** lab12;

**var**

i, j, sum, max : integer;

arr : **Array** [1..3, 1..3] **of** integer;

**Begin**

sum := 0;

max := 0;

**for** i := 1 **to** 3 **do**

**for** j := 1 **to** 3 **do**

**begin**

arr[i,j] := j;

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

**if** arr[i,j] > max **then**

max := arr[i,j];

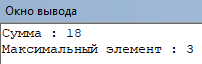
**end**;

writeln ('Сумма : ', sum);

writeln ('Максимальный элемент : ', max );

**End**.

**9.1)**



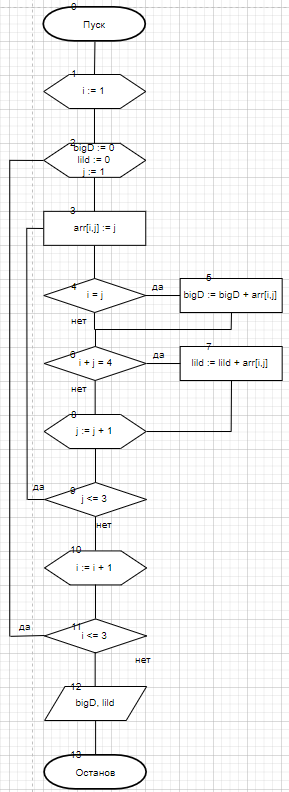
**10.1)** Задача была решена с помощью циклов конструкции for и применением оператора условия If then. Перед началом работы программы проходит инициализация переменных “sum”, “max”. Далее мы попадаем во внешний цикл от параметра i, после чего сразу же во внутренний по переменной j, где одновременно заполняется массив arr, каждый элемент суммируется в переменной “sum” и с помощью логического уравнения “arr[i,j] > max” все элементы массива проверяются на величину, чтобы найти максимальное значение, когда циклы завершают работу, а то есть, когда во внешнем цикле не выполнится условие i <= 3, а во внутреннем j <= 3 , программа выводит сумму элементов и максимальное значение.

**Задача 2:**

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

**5.2)** bigD := bigD + arr[i,j], lild := lild + arr[i,j], i := i + 1, j := j + 1

**6.2)**



**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **i** | **Параметр внешнего цикла** | **int** |
| **j** | **Параметр внутреннего цикла** | **int** |
| **bigD** | **Сумма элементов главной диагонали массива arr** | **int** |
| **lild** | **Сумма элементов побочной диагонали массива arr** | **int** |
| **arr** | **Двумерный массив размерностью 3х3** | **int** |

**8.2)**

**program** lab12;

**var**

i, j, bigD, lild : integer;

arr : **Array** [1..3, 1..3] **of** integer;

**Begin**

bigD := 0;

lild := 0;

**for** i := 1 **to** 3 **do**

**for** j := 1 **to** 3 **do**

**begin**

arr[i,j] := j;

**if** i = j **then**

bigD := bigD + arr[i,j];

**if** i+j = 4 **then**

lild := lild + arr[i,j];

**end**;

writeln ('Сумма по главной диагонали : ', bigD);

writeln ('Сумма по побочной диагонали : ', lild );

**End**.

**9.2)**



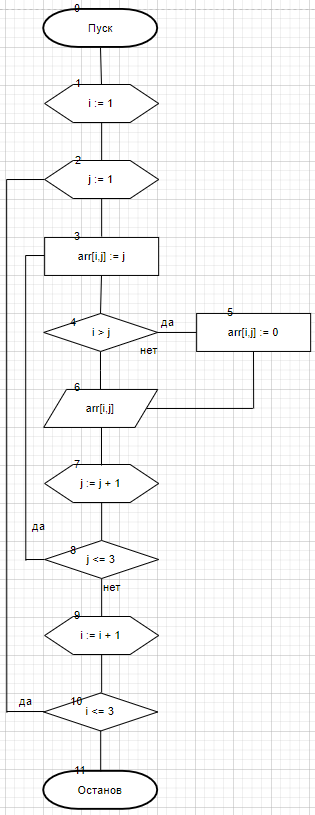
**10.2)** При решении задачи использовался оператор условия If then, и был применен вложенный цикл конструкции for. В начале работы программы инициализируются переменные “bigD”, “lild”. После инициализации переменных программа попадает во внешний цикл for с параметром i, после чего сразу начинается работа внутреннего цикла for с параметром j, где одновременно заполняется массив и вычисляется сумма элементов каждой диагонали. Для главной диагонали логическое условие – i = j , при истинности переменная “bigD” увеличивается на данный элемент, иначе проверяется следующее условие - i+j = 4 , если истина, то уже переменная “lild” увеличивается на данный элемент. Циклы завершают работу при неистинности условий для внешнего цикла - i<=3, а для внутреннего j<=3. Как завершаются циклы, программа выводит сумму главной диагонали и сумму побочной.

**Задание 3:**

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

**5.3)** i := i + 1, j := j + 1

**6.3)**



**7.3)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **i** | **Параметр внешнего цикла** | **int** |
| **j** | **Параметр внутреннего цикла** | **int** |
| **arr** | **Двумерный массив размерностью 3х3** | **int** |

**8.3)**

**program** lab12;

**var**

i, j : integer;

arr : **Array** [1..3, 1..3] **of** integer;

**Begin**

writeln ('Массив arr : ');

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 3 **do**

**begin**

arr[i,j] := j;

**if** i > j **then**

arr[i,j] := 0;

write (arr[i,j]);

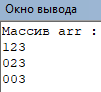
**end**;

writeln();

**end**

**End**.

**9.3)**



**10.3)** Задача была решена с помощью циклов конструкции for, также был использован оператор условия If then. С начала работы программы мы попадаем во внешний цикл от параметра i, после чего сразу же во внутренний по переменной j, где одновременно заполняется массив arr, дальше проверяется условие i < j, если истина, то данному элементу присваивается значение нуля, в любом другом исходе данный элемент выводится на экран, когда циклы завершают работу, а то есть когда во внешнем цикле не выполнится условие i <= 3, а во внутреннем j <= 3 .

**Задача 4:**

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

**7.4)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |  |  |  |
| **i** | **Параметр внешнего цикла** | **int** |  |  |  |
| **j** | **Параметр внутреннего цикла** | **int** |  |  |  |
| **arr** | **Двумерный массив размерностью 3х4** | **int** |  |  |  |
| **l** | **Параметр цикла.** | **int** |  |  |  |
| **sum** | **Накопительная переменная, сумма элементов некой строки** | **int** |  |  |  |
| **b** | **Промежуточная переменная.** | **int** |  |  |  |

**8.4)**

**program** lab10;

**uses** crt;

**var**

i, j, l, sum, b: integer;

arr : **Array**[1..3,1..4] **of** integer;

**Begin**

randomize;

**for** i := 1 **to** 3 **do**

**begin**

sum := 0;

**for** j := 1 **to** 3 **do**

**begin**

arr[i,j] := random(4);;

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

**end**;

arr[i,4] := sum;

**end**;

writeln('Массив без сортировки : ');

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 4 **do**

write(arr[i,j]);

writeln;

**end**;

**for** i := 1 **to** 2 **do**

**for** l := i+1 **to** 3 **do**

**if** arr[i,4] > arr[l,4] **then**

**for** j := 1 **to** 4 **do**

**begin**

b := arr[i,j];

arr[i,j] := arr[l,j];

arr[l,j] := b;

**end**;

writeln('Массив после сортировки : ');

**for** i:=1 **to** 3 **do**

**begin**

**for** j:=1 **to** 3 **do**

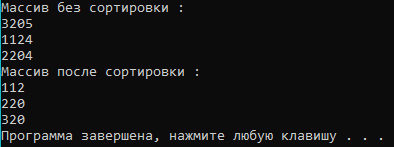
write(arr[i,j]);

writeln;

**end**;

**End**.

**9.4)**



**10.4)** Задача была решена с помощью циклов конструкции for, также применялся оператор условия If then. С начала работы программы мы попадаем во внешний цикл от параметра i, в котором сначала обнуляется переменная “sum”, после чего сразу же начинается внутренний цикл по переменной j, где одновременно заполняется массив arr, и считается сумма данной строки, после выхода из внутреннего цикла сумма записывается в четвертый столбец матрицы. Выход из внешнего цикла, при невыполнении условия i<=3. Дальше выводится неотсортированная матрица. Следующим шагом начинается внешний цикл по переменной “i”, в котором вложенный цикл по переменной “j”, в них проходит сортировка строк массива по возрастанию, после сортировки выводится конечная матрица с уже отсортированными строками по сумме.

**11)** Мы научились реализовывать алгоритмы многоступенчатых циклических вычислительных процессов средствами компилятора free Pascal.