**Лабораторная работа №10**

**Итерационные ЦВП с управлением по индексу и функции.  
Вариационный ряд.**

**Цель:** научиться итерационным циклическим процессам с управлением по индексу и функции средствами компилятора free Pascal.

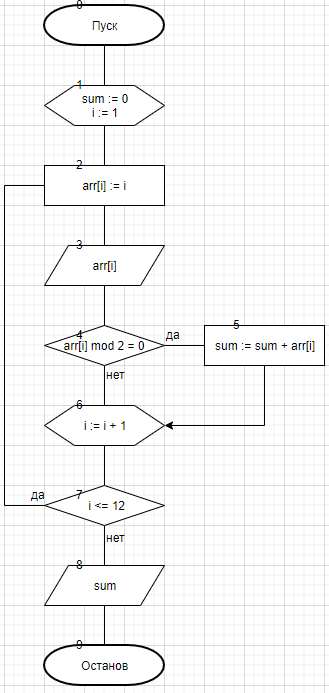
**Используемое оборудование:** пк, PascalABC.

**Задание 1:**

**4.1)** Дан одномерный массив. Найти сумму четных ( по значению ) элементов массива.

**5.1) arr[i] mod 2, sum = sum + arr[i]**

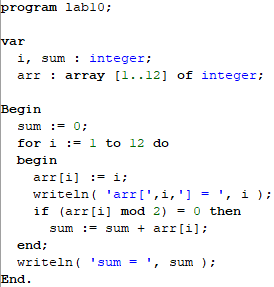
**6.1)**



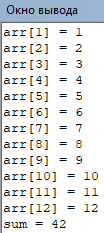
**7.1)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **i** | **Переменная, параметр цикла, индекс массива arr** | **int** |
| **sum** | **Переменная, сумма четных элементов массива arr** | **int** |
| **arr** | **Массив значений размером 12 ячеек** | **int** |

**8.1)**



**9.1)**



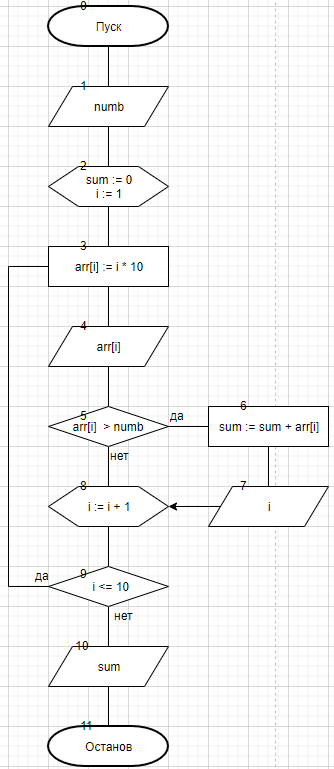
**10.1)** Для решения поставленной задачи мы воспользовались циклом конструкции for и массивом типа integer размерностью 12 элементов, а также логической конструкцией if then. С начала работы программы происходит инициализация переменных суммы и параметра цикла, затем в цикле присваивается значение параметра элементу массива индексом, соответствующего параметру цикла. Выводится элемент массива, дальше программа переходит к логической конструкции, условие которой arr[i] mod 2 = 0, если истина, то выполняется увеличение суммы на значение данного элемента массива, после чего параметр цикла увеличивается на единицу, иначе при не истине изменяется только параметр цикла на единицу. Цикл работает, пока выполняется условие i <= 12. При окончании работы цикла выводится сумма четных по значению элементов массива arr.

**Задание 2:**

**4.2) Дан одномерный массив. С клавиатуры вводится число. Найти сумму всех элементов массива, значения которых больше числа, введенного с клавиатуры, и вывести их индексы.**

**5.2) i\*10, sum = sum + arr[i]**

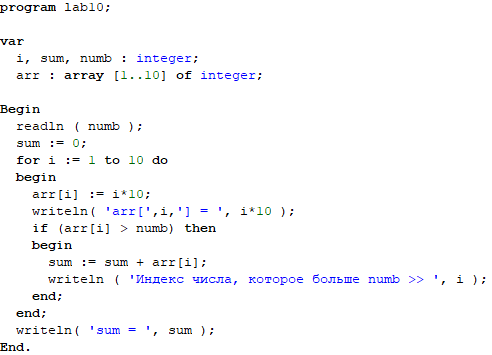
**6.2)**



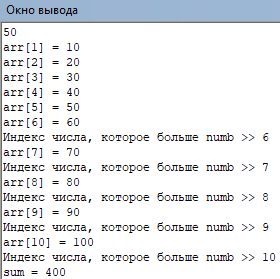
**7.2)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **i** | **Переменная, параметр цикла, индекс массива arr** | **int** |
| **sum** | **Переменная, сумма элементов массива arr, которые больше значения numb** | **int** |
| **arr** | **Массив значений размером 12 ячеек** | **int** |
| **numb** | **Переменная, вводимое число с клавиатуры.** | **int** |

**8.2)**



**9.2)**



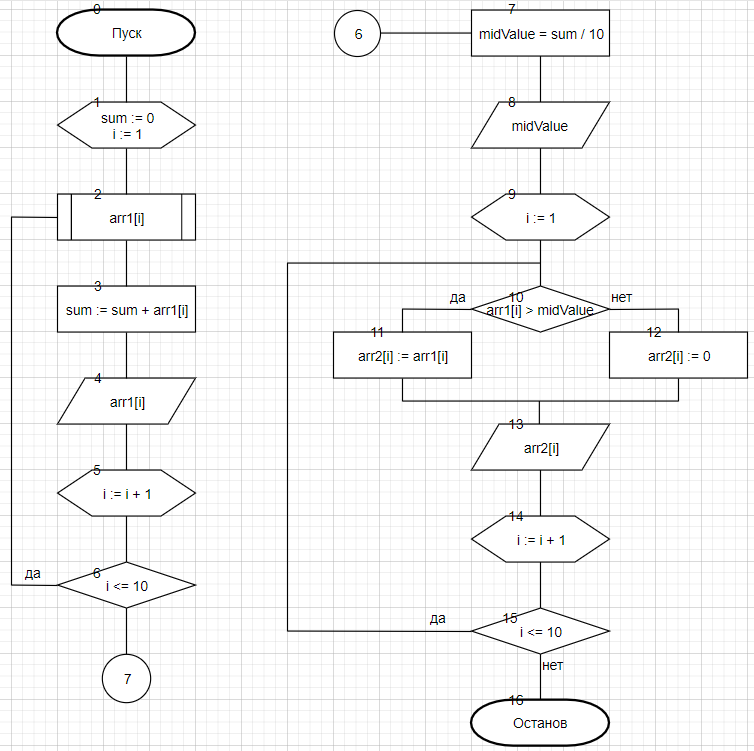
**10.2)** Для решения поставленной задачи мы воспользовались циклом конструкции for и массивом типа integer размерностью 10 элементов, а также логической конструкцией if then. С начала работы программы выполняется ввод значения numb, после инициализация переменных суммы и параметра цикла. Когда программа попадает в цикл, в нем присваивается значение параметра, умноженного на десять, элементу массива индексом, соответствующего параметру цикла. Следующей операцией выводится значение элемента массива, дальше программа переходит к логической конструкции, условие которой arr[i] > numb, если истина, то выполняется увеличение суммы на значение данного элемента массива и вывод индекса этого элемента, после чего параметр цикла увеличивается на единицу, иначе при не истине изменяется только параметр цикла на единицу. Цикл работает, пока выполняется условие i <= 10. При окончании работы цикла выводится сумма элементов массива “arr” больших значения numb.

**Задание 3:**

4.3) Дан одномерный массив. Найти его среднее арифметическое. Составить второй массив, элементами которого будут элементы первого массива, которые больше среднего арифметического. Остальные элементы заменить нулями.

**5.3)** sum = sum + arr[i], midValue = sum/10

**6.3)**



**7.3)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| **sum** | **Переменная, сумма элементов массива arr1** | **int** |
| **i** | **Переменная, параметр цикла** | **int** |
| **midValue** | **Переменная, ср. арифметическое значение элементов массива arr1** | **real** |
| **arr1, arr2** | **Массивы размерностью 10** | **int** |

**8.3)**

**program** lab10;

**uses** crt;

**var**

sum, i : integer;

midValue : real;

arr1, arr2 : **array** [1..10] **of** integer;

**Begin**

randomize;

sum := 0;

write( 'Первый массив : ' );

**for** i := 1 **to** 10 **do**

**begin**

arr1[i] := random(101);

sum := sum + arr1[i];

write( arr1[i], ' ' );

**end**;

midValue := sum / 10;

writeln();

writeln('Ср. арифметическое первого массива : ', midValue);

write( 'Второй массив : ' );

**for** i := 1 **to** 10 **do**

**begin**

**if** ( arr1[i] > midValue ) **then**

arr2[i] := arr1[i]

**else**

arr2[i] := 0;

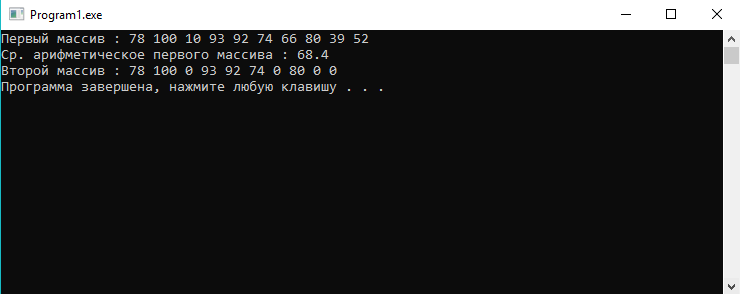
write( arr2[i], ' ' );

**end**;

writeln();

**End**.

**9.3)**



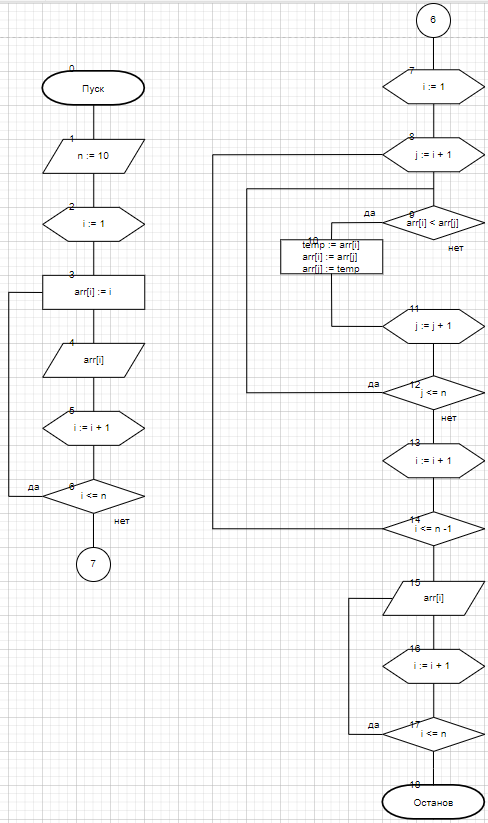
**10.3)** Чтобы решить поставленную задачу, использовались массивы, циклы конструкции for, а также логическая конструкция if then else. В начале работы программы инициализируется переменная “sum”, дальше начинается цикл с параметром “i”, где заполняется массив arr1, каждый элемент которого суммируется в переменной “sum”, работа цикла продолжается пока истинно условие i <= 10. Последующим шагом вычисляется среднее значение “midValue” по формуле sum/10, после этого начинается еще один цикл, проверяющий первым действием условие arr1[i] > midValue, при истинности которого элементу массива arr2[i] присваивается элемент массива arr1[i], иначе присваивается ноль. При окончании работы цикла, программа завершает работу.

**Задание 4:**

**4.4)** Дан одномерный массив. Упорядочить его по убыванию.

**5.4)-**

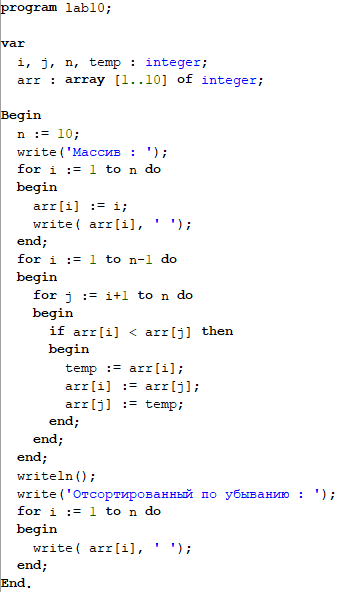
**6.4)**



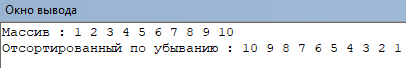
**7.4)**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i | Переменная, параметр цикла | Int |
| j | Переменная, параметр цикла | Int |
| n | Переменная, кол-во элементов массива | Int |
| temp | Промежуточная переменная, сохраняет в себе значение некого элемента массива | Int |
| arr | Массив значений | Int |

**8.4)**



**9.4)**



**10.4)**Для решения поставленной задачи был использован метод сортировки пузырьком, также применялись циклы конструкции for, и воспользовались логической конструкцией if then. С начала работы программы инициализируется переменная n и заполняется значениями массив arr, после чего программа попадает в цикл for по параметру i, условие которого i <= n-1. В этом же цикле используется еще один по параметру j = i + 1 с условием j <= n, где проверяется условие arr[i] < arr[j], при истинности которого сравниваемые элементы меняются местами, иначе проверяется условие второго цикла, и в зависимости от этого цикл продолжает работу или нет. При завершении работы внутреннего цикла, параметр внешнего цикла увеличивается на единицу, и в зависимости от условия продолжается его работа или нет. При завершении работы внешнего цикла, выводится отсортированный массив с помощью цикла for параметра i, и программа завершает свою работу.

**11)** Мы научились реализовывать алгоритмы итерационных ЦВП с управлением по индексу и функции средствами компилятора free Pascal.