Лабораторная работа №10

**Тема:** Разветвляющиеся вычислительные процессы

**Цель:** Научиться реализовывать алгоритмы на разветвляющиеся вычислительные процессы

**Оборудование:** ПК, PascalABC.NET

Задача №1

**Постановка задачи:** Дан одномерный массив. Найти сумму четных (по значению) элементов массива.

**Математическая модель:** Для всех элементов массива в цикле будем проверять условие: «a[i] mod 2 = 0». При прохождении условия будем прибавлять к переменной «s» значение элемента.

**Блок-схема:**

C:\Users\Denis\Documents\дз\Информатика\ЛБ10, РВП продолжение\Блок-схемы\1.png

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| A[1..n] | массив | integer |
| n | количество элементов | константа |
| s | сумма четных элементов | integer |
| i | счетчик | integer |

**Код программы:**

**const**

n = 10;

**var**

a: **array** [0..n] **of** integer;

i, s: integer;

**begin**

**for** i := 0 **to** n **do** a[i] := random(15);

writeln(A);

s:=0;

**for** i := 0 **to** n **do**

**if** i **mod** 2 = 0

**then**

s := s + A[i];

writeln(s);

**end**.

**Результат выполненной работы:**



**Анализ результатов вычисления:**

Так как нумерация элементов массива начинается с «0» и ограничивается константой «n» программа будет складывать числа начиная с первого (так как 0 – четное число)

Задача №2

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

**Математическая модель:** Для всех элементов массива в цикле будем проверять условие: «a[i] > x». При прохождении условия будем прибавлять к переменной «s» значение элемента и выводить индекс.

**Блок-схема:**

C:\Users\Denis\Documents\дз\Информатика\ЛБ10, РВП продолжение\Блок-схемы\2.png

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| A[1..10] | массив | integer |
| s | сумма элементов больших чем x | integer |
| x | вводимое число | integer |
| i | счетчик | integer |
| n | количество элементов | константа |

**Код программы:**

**const**

n = 10;

**var**

A: **array** [0..n] **of** integer;

i, x, s: integer;

**begin**

**for** i := 0 **to** n **do** a[i] := random(15);

writeln(A);

readln(x);

s:=0;

**for** i := 0 **to** n **do**

**if** A[i] > x **then**

**begin**

s := s + A[i];

write(i, ' ');

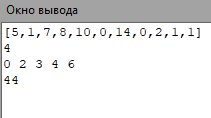
**end**;

writeln('');

writeln(s);

**end**.

**Результат выполненной работы:**



**Анализ результатов вычисления:**

В данной задаче нумерация элементов также начинается с «0», это стоит учесть при анализе выведенных данных

Задание №3

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

**Математическая модель:** Сначала высчитаем среднее арифметическое, сложив в цикле все элементы массива и поделив их на количество, а затем, в другом цикле будем сравнивать среднее арифметическое и элементы массива А, и исходя из результатов сравнения заполнять массив B

**Блок-схема:**

**C:\Users\Denis\Documents\дз\Информатика\ЛБ10, РВП продолжение\Блок-схемы\3.png**

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| A[1..n],B[1..n] | массивы | integer |
| i | счетчик | integer |
| n | количество элементов | константа |
| S | среднее арифметическое | real |

**Код программы:**

**const**

n = 10;

**var**

A: **array** [0..n] **of** integer;

B: **array** [0..n] **of** integer;

i: integer;

s: real;

**begin**

s:=0;

**for** i := 0 **to** n **do**

**begin**

a[i] := random(15);

s := s + A[i];

**end**;

s := s / n;

writeln(A);

writeln(s);

**for** i := 0 **to** n **do**

**if** A[i] > s **then**

B[i] := A[i]

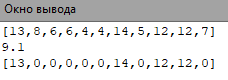
**else**

B[i] := 0;

writeln(B);

**end**.

**Результат выполненной работы:**



**Анализ результатов вычисления:**

Стоит учесть что значение среднего арифметического должно содержаться в вещественной переменной. Нумерация элементов в массиве начинается с «0».

Задание №4

**Постановка задачи:** Дан одномерный массив. Упорядочить его по убыванию

**Математическая модель:** Для того чтобы упорядочить массив по убыванию прибегнем к циклу в цикле. В верхнем цикле мы будем идти по всем элементам, начиная с первого и заменять его (в случае если он меньше) на найденный через нижний цикл наименьший элемент среди оставшихся.

**Блок – схема:**

**C:\Users\Denis\Documents\дз\Информатика\ЛБ10, РВП продолжение\Блок-схемы\4.png**

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| A[0..n] | массив | integer |
| n | количество элементов массива | константа |
| i,j | счетчики | integer |
| max,t | промежуточные переменные | integer |

**Код программы:**

**const**

n = 10;

**var**

a: **array** [0..n] **of** integer;

i, j, t, k, max: integer;

**begin**

**for** i := 0 **to** n **do** a[i] := random(150);

writeln(A);

**for** i := 0 **to** n **do**

**begin**

max := a[i];

**for** j := i + 1 **to** n **do**

**if** a[j] > max **then**

**begin**

max := a[j];

k := j;

**end**;

**if** a[k] > a[i]

**then**

**begin**

t := a[i];

a[i] := a[k];

a[k] := t;

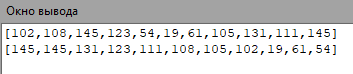
**end**;

**end**;

writeln(A);

**end**.

**Результат выполненной работы:**



**Анализ результатов вычисления:**

Нужно внимательно следить за проверкой условия и вложенными циклами. Расставлять операторные скобки когда в цикле или условии несколько действий.

**Вывод:**

Средствами PascalABC.NET были реализованы алгоритмы на разветвляющиеся вычислительные процессы