Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

|  |
| --- |
|  |
|  |

# оТЧЕТ

по лабораторной работе

на тему:

Одномерные массивы

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 851002 |  | К. Д. Цыбулько |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2018

1. **Теоретические сведения по теме лабораторной работы**

*Массив* – упорядоченная совокупность однотипных элементов, имеющих

общее имя. Данное имя называется полной переменной , так как ее значением

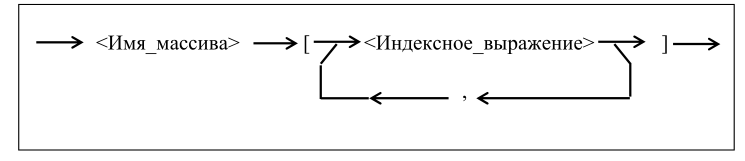
является весь массив.

*Тип элементов* массива называется базовым типом массива .

Для выделения отдельных элементов массива используются *индексы*

(индексные выражения). Индекс задает правило вычисления номера нужного

элемента массива.

****Для указания определенного элемента массива используется формат

*Индексное выражение* – это выражение скалярного типа. Количество

индексных выражений определяется количеством измерений массива.

**Задание на лабораторную работу**

* 1. **Постановка задачи**

**Задание №36**

Задан числовой массив . Найти отрезок массива максимальной длины, в котором первое число равно последнему, второе предпоследнему и т.д.

* 1. **Ожидаемый результат**

1 3 1 3 0 3 2 4 1 3 3 3 1 3 1 0 2 2 4 1

Longest segment

1 3 3 3 1

Length 5

1. **Выполнение**
   1. **Разработка алгоритма**

*Таблица 2.1 используемые идентификаторы*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения |
| B | Массив | TMAS | Генерируется случайным образом | - |
| i | Параметр цикла левой границы | integer | 1 | i:=i+1 |
| j | Параметр цикла правой границы | integer | n | j:=j-1 |
| bord1 | Левая граница максимального палиндрома | integer | i | bord1:=i |
| bord2 | Правая граница максимального палиндрома | integer | j | bord2:=j |
| Length | Длина участка массива с которым ведется работа в данный момент | integer | j-i+1 | Length:=j-i+1 |
| max | Длина максимального палиндрома | integer | 0 | Max:=Length |
| k | Параметр цикла вывода максимального палиндрома | integer | bord1 | k:=k+1 |

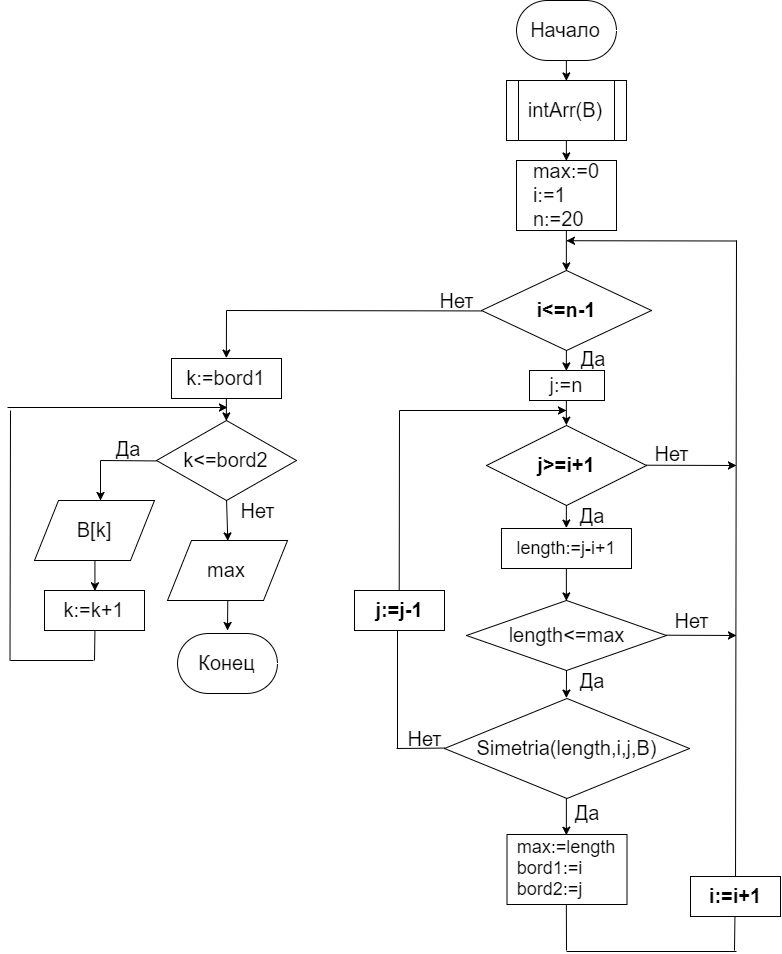


Рисунок 2.1 – Схема работы программы (Main)

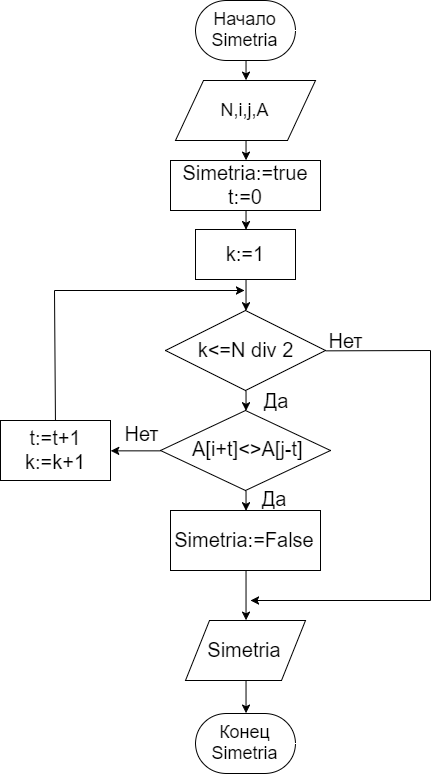


Рисунок 2.2 – схема работы функции (Simetria)

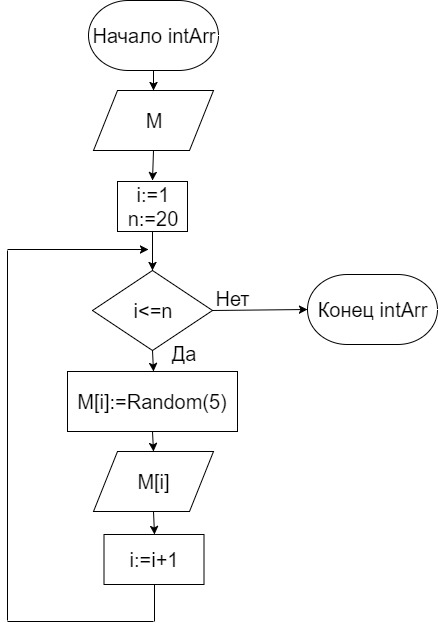


Рисунок 2.3 – схема работы процедуры (intArr)

1. Ведем проверку по двум циклам (правая и левая границы). Первым делом берем на проверку отрезок, равный массиву.
2. Далее сдвигаем правую границу и при каждом сдвиге проверяем на палиндром, как только палиндром найден – выходим и изменяем левую границу на 2, а правую на n-1 и тд. При переходе на новый отрезок проверяем, не короче ли он максимального палиндрома (найденного на данный момент), если короче, то меняем левую границу.
3. Как только найден палиндром, проверяем не короче ли он предыдущего и записываем его координаты и длину
4. В конце выводи сам максимальный палиндром и его максимальное значение
   1. **Текст программы и его описание**

**program** LR4;

**uses**

SysUtils;

{$APPTYPE CONSOLE}

**const** n=20; //number of characters

**type**

TMAS=array[1..n] of integer;

**procedure** intArr( var M:TMAS); //array definition

**var** i:Integer;

**begin**

Randomize;

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

**begin**

M[i]:=Random(5);

write(M[i]:2);

**end;**

Writeln;

**end;**

**function** Simetria(N,i,j:Integer; A:TMAS):boolean; //determine whether the array is symmetric about its midpoint

**var** k,t:Integer;

**begin**

t:=0;

Simetria:=true;

**for** k:=1 to N div 2 do

**begin**

**if** A[i+t]<>A[j-t] then

**begin**

Simetria:=False;

break;

**end;**

t:=t+1;

**end;**

**end;**

**var** B :TMAS;

i,j,bord1,bord2,Length,max :Integer;

simm :boolean;

**BEGIN**

intArr(B);

max:=0;

**for** i:=1 to n-1 do //first board

**for** j:=n downto i+1 do //second board

**begin**

length:=j-i+1;

**if** length<=max **then** Break;

**if** Simetria(length,i,j,B) **then** //change of the maximum polindrome value

**begin**

max:=length;

bord1:=i;

bord2:=j;

break;

**end;**

**end;**

writeln('Longest segment');

**for** k:=bord1 to bord2 **do**

write(B[k],' ');

writeln;

writeln('Length ',max);

readln;

END.

**3.3 Тестирование и отладка программы**

*Таблица 2.2 Прохождение тестов программы*

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| 1 | 0  2  2  4  3  2  0  4  0  4  0  1  4  0  4  1  1  2  4  1 | 0 4 0 4 0  Length 5 | Тест пройден |
| 2 | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | 0  Length 0 | Тест пройден |
| 3 | 1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0 | 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1  Length 19 | Тест пройден |