Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Севастопольский государственный университет»

Кафедра ИС

Отчет

По лабораторной работе №3

«ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ОДНОМЕРНЫХ МАССИВОВ»

По дисциплине “Алгоритмизация и программирование”

Вариант №3

Выполнил:

ст. гр. ИС-12

Волобуев Ю.С.

Проверил:

Забаштанский А.К.

Севастополь

2017

1 ЦЕЛЬ РАБОТЫ

Изучить способы представления массивов в памяти ЭВМ, получить практические навыки реализации алгоритмов обработки одномерных массивов, исследовать свойства алгоритма сортировки.

2 ПОСТАНОВКА ЗДАЧИ

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1) Произведение элементов массива с четными номерами;

2) Сумму элементов массива, расположенных между первым и последним нулевыми элементами.

3) Преобразовать массив таким образом, чтобы сначала располагались все неотрицательные элементы, а потом все отрицательные.

4) Выполнить сортировку каждой части массива методом прямого выбора.

3 СТРУКТУРНАЯ СХЕМА АЛГОРИТМА

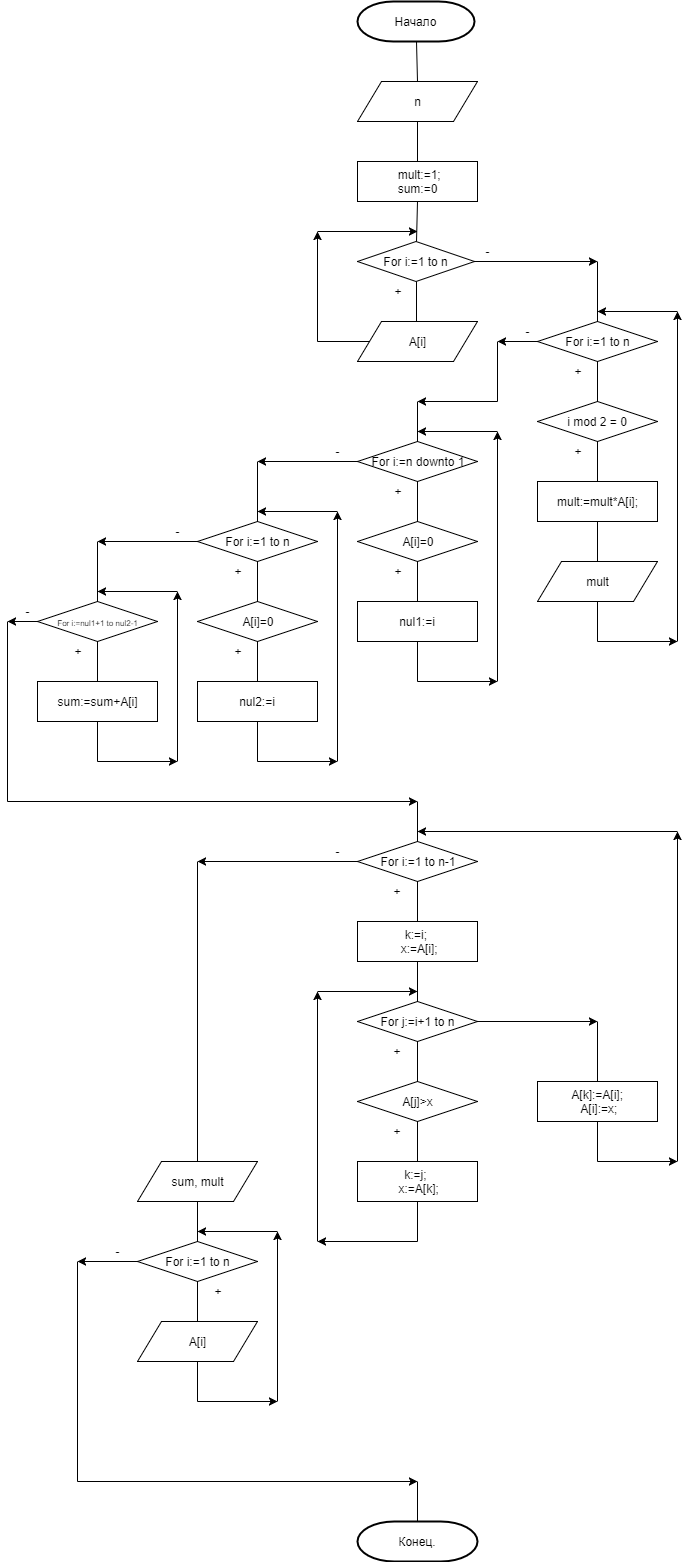


Рисунок 1 - Структурная схема алгоритма

4 ТЕКСТ ПРОГРАММЫ

**Program** LAB4;

**Const** n=5; m=7;

**Type** matrix=**array** [1..n, 1..m] **of** integer;

**Var** A: matrix;

s, p, num, numcon: integer;

**Procedure** Input (**Var** A: matrix);

**Var** i,j: integer;

**Begin**

Writeln ('Введите значения элементов массива:');

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

**Begin**

**For** j:=1 **to** m **do**

**Begin**

Readln(A[i,j]);

**End**;

Writeln('---');

**End**;

**End**;

**Procedure** Poisk1 (A: matrix; **Var** s: integer);

**Var** i, j: integer;

**Begin**

s:=0;

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

**For** j:=1 **to** m **do**

**If** (A[i,j]=0) **then**

**Begin**

s:=s+1;

**Break**;

**End**;

**End**;

**Procedure** Poisk2 (A:matrix; **Var** p, num, numcon:integer);

**Var** i, j: integer;

**Begin**

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

**Begin**

num:=0;

**For** j:=1 **to** m **do**

**If** (a[i,j+1]<m) **and** (A[i,j]=A[i][j+1]) **then** num:=num+1;

**If** (num>numcon) **then**

**Begin**

numcon:=num;

p:=i;

**End**;

**End**;

**End**;

**Procedure** Output (**Const** A: matrix; **Const** s, p, numcon:integer);

**Var** i, j: integer;

**Begin**

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

**Begin**

**For** j:=1 **to** m **do**

Write(A[i,j], ' ' );

Writeln;

**End**;

Writeln('Количество столбцов хоть с одним нулевым элементом: ', s);

Writeln('Номер строки с самой длинной серией одинаковых элементов: ',p);

Writeln('Количество одинаковых элементов: ', numcon);

**End**;

**Begin**

Input(A);

Poisk1(A, s);

Poisk2(A, p, num, numcon);

Output(A, s, p, numcon);

**End**.

5 ТЕСТОВЫЕ ПРИМЕРЫ

В случае, когда нулевые элементы находятся под разными индексами, вычисления проходят верно, единственное, что если первый четный элемент равен нулю, то значение произведения зануляется.

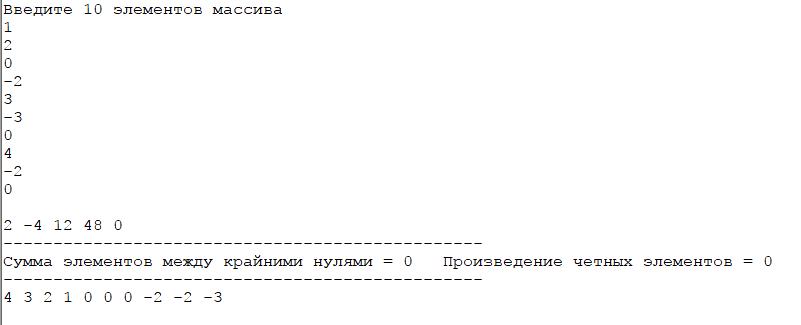


Рисунок 2 – Тестовый пример для нескольких нулей

При общей границе (единственный нулевой элемент в массиве) вычисления так же проходят верно.

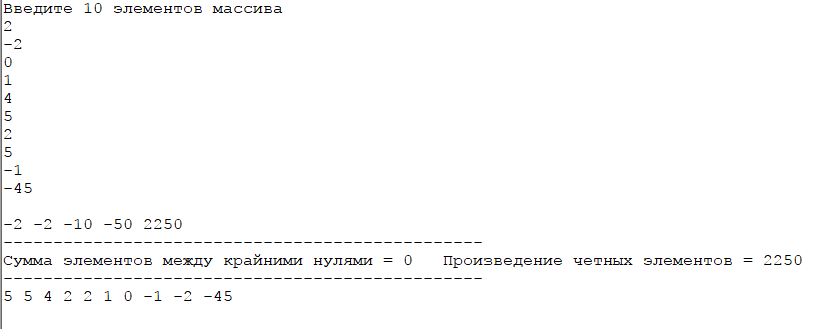


Рисунок 3 – Тестовый пример для совпадающих границ вычислений

ВЫВОДЫ

Был написан текст программы, попутно были изучены способы обработки и сортировки массива. Из всех методов сортировки самый эффективный – сортировка “Пузырьком”. Составлены тестовые примеры соответственно варианту из методички. Программа работает верно, все вычисления производятся корректно. Массив был отсортирован методом прямого выбора. Получены навыки работы с массивами и улучшены навыки работы с циклами.