Министерство образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

ОТЧЕТ

По лабораторной работе №7

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ОДНОМЕРНЫХ МАССИВОВ

Выполнил: ст. гр. ИТб-11

Куркчи А. Э.

Проверил:

Забаштанский А.К.

Севастополь

2014

1. Цель работы

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

2. Постановка задач

Вариант №2

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

1. сумму отрицательных элементов массива;
2. произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию, используя алгоритм сортировки методом вставки.

3. Структурная схема

На рисунке 1 изображена схема алгоритма работы программы.

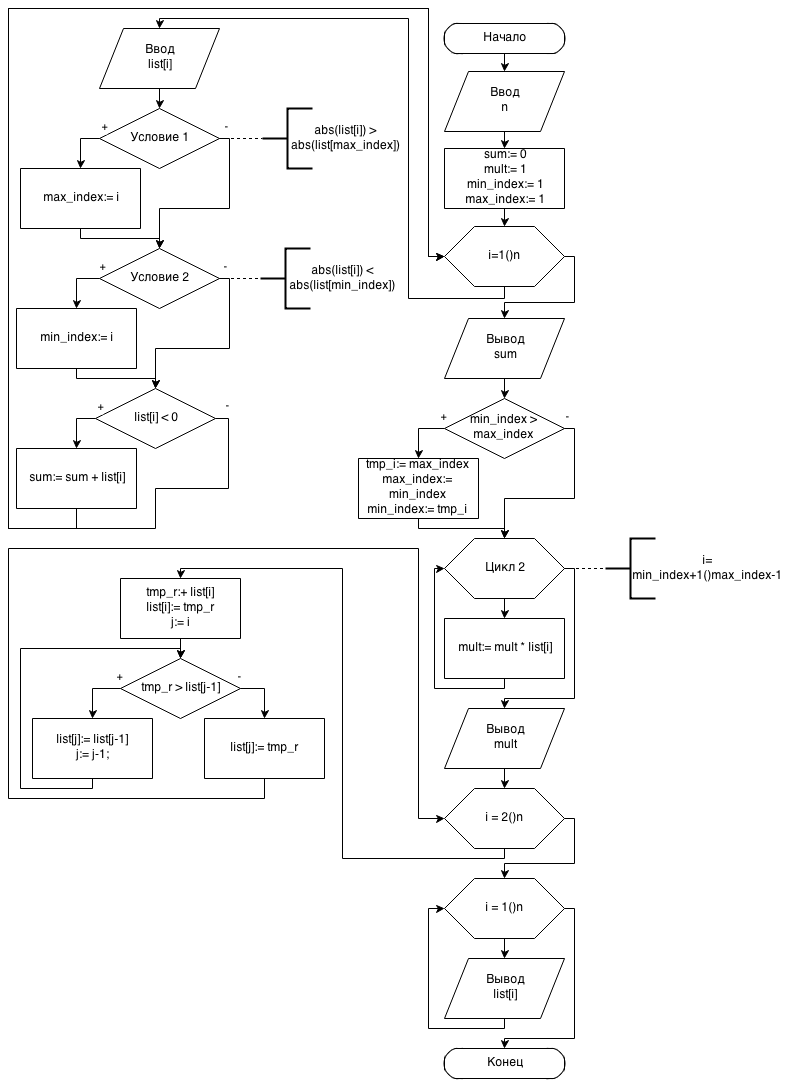
****

Рисунок 1 – структурная схема

4. Текст программы

**program** lab7;

**uses** crt;

**var** list:**array**[0..1000] **of** real;

n,i,j,min\_index,max\_index,tmp\_i:integer;

sum,mult,tmp\_r:real;

**begin**

sum:= 0.0;

mult:= 1.0;

writeln('Введите кол-во элементов n');

readln(n);

writeln('Введите ',n,' элементов массива');

min\_index:= 1;

max\_index:= 1;

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

read(list[i]);

**if**(abs(list[i]) > abs(list[max\_index])) **then**

max\_index:= i; {Индекс максимального элемента}

**if**(abs(list[i]) < abs(list[min\_index])) **then**

min\_index:= i; {Индекс минимального элемента}

**if**(list[i] < 0) **then**

sum:= sum + list[i]; {Считаем сумму}

**end**;

writeln('Сумма отрицательных элементов массива:');

writeln(sum);

**if**(min\_index > max\_index) **then begin**

tmp\_i:= max\_index;

max\_index:= min\_index;

min\_index:= tmp\_i;

**end**;

**for** i:=min\_index+1 **to** max\_index-1 **do**

mult:= mult \* list[i]; {Считаем произведение}

writeln('Произведение элементов между максимальным и минимальным:');

writeln(mult);

**for** i:= 2 **to** n **do begin** {Сортировка методом вставки}

tmp\_r:= list[i];

list[0]:= tmp\_r;

j:= i;

**while**(tmp\_r > list[j-1]) **do begin**

list[j]:= list[j-1];

j:= j-1;

**end**;

list[j]:= tmp\_r;

**end**;

writeln('Отсортированые методом вставки элементы:');

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

write(list[i],' ');

writeln();

readln;

**end**.

5. Тестовые примеры

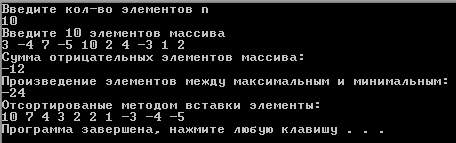


Рисунок 2 – пример №1

На рисунке 2 изображена работа программы с массивом из 10 элементов, максимальный по модулю 5й элемент «10», минимальный по модулю 9й элемент «1». Между максимальным и минимальным 3 элемента: «2», «4», «-3» их произведение равно «-24». Отрицательных элементов 3: «-4», «-5», «-3» их сумма равна «-12».

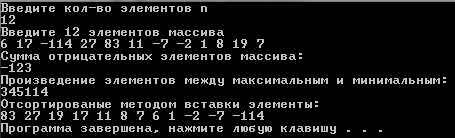


Рисунок 3 – пример №2

На рисунке 2 изображена работа программы с массивом из 12 элементов, максимальный по модулю 3й элемент «-114», минимальный по модулю 9й элемент «1». Между максимальным и минимальным 5 элементов: «27», «83» , «11», «-7», «-2» их произведение равно «345114». Отрицательных элементов 3: «-114», «-7», «-2» их сумма равна «-123».

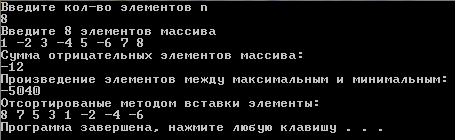


Рисунок 4 – пример №3

На рисунке 2 изображена работа программы с массивом из 8 элементов, максимальный по модулю 8й элемент «8», минимальный по модулю 1й элемент «1». Между максимальным и минимальным 6 элементов: «-2», «3» , «-4», «5», «-6», «7» их произведение равно «-5040». Отрицательных элементов 3: «-2», «-4», «-6» их сумма равна «-12».

Вывод

В ходе лабораторной работы были изучены способы представления массивов в памяти ЭВМ, получены практические навыки реализации алгоритмов, обработки одномерных массивов, исследованы свойства алгоритма сортировки. Было выявлено, что количество операций сравнения при сортировке массива прямо пропорционально количеству элементов в массиве, в то время как количество перестановок зависит от начального порядка элементов в массиве.