**ЛАБОРАТОРНАЯ РАБОТА №5**

**«ОБРАБОТКА ОДНОМЕРНЫХ ДИНАМИЧЕСКИХ МАССИВОВ С ПОМОЩЬЮ ФУНКЦИЙ»**

**Цель работы**

Изучение особенностей представления и обработки одномерных массивов в языках С/С++ с учетом связи указателей и массивов. Получение практических навыков реализации алгоритмов обработки одномерных динамических массивов средствами языков С/С++. Исследование особенностей обработки одномерных динамических массивов.

**Задания**

1. Написать программу, решающую поставленную задачу (вариант 12 методических указаний):

Дана последовательность, состоящая из вещественных чисел. Необходимо найти такое ai, что выполнится следующее: max[cos(a1), cos(a2) … cos(an)] = cos(ai). Упорядочить элементы последовательности ap, ap+1, …, aq по возрастанию, используя алгоритм сортировки методом прямого выбора.

Оформить программу в виде функций, выполняющих выделение памяти и заполнение динамического массива с клавиатуры, обработку и сортировку массива согласно варианту задания, вывод массива на экран.

1. Разработать структурные схемы алгоритма для всех функций решения задачи;
2. Разработать тестовые примеры и выполнить тестирование и отладку написанной программы;

**Текст программы**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

**int** i, j; //глобальные параметры

//функция выделения памяти и ввода массива

**float**\* input\_a(**float** \*a, **int** n)

{

printf("Enter the array:\n");

a = (**float**\*)malloc(**sizeof**(**float**)\*n); //выделение памяти под массив

**for** (i=0;i<n;i++) //заполнение массива

scanf("%f",&a[i]);

**return**(a);

}

//функция нахождения элемента, дающего максимальный косинус

**void** max\_cos(**float** a[], **int** n)

{

**int** max\_i=0; //позиция нужного элемента

**for** (i=1;i<n;i++) //поиск нужного элемента

**if** (cos(a[i])>cos(a[max\_i]))

max\_i=i;

**return**(a[max\_i]); //вывод ответа

}

//функция сортировки массива в пределах [ap..aq]

**void** sort(**float** a[], **int** n, **int** p, **int** q)

{

**int** i\_min, //позиция минимального элемента

temp; //переменная для обмена

**for** (i=p-1;i<q-1;i++) //сортировка методом прямого выбора

{

i\_min=i;

**for** (j=i+1;j<q;j++)

**if** (a[j]<a[i\_min])

i\_min=j;

temp=a[i];

a[i]=a[i\_min];

a[i\_min]=temp;

}

printf("array sorted\n\n");

}

//функция вывода массива

**void** output\_a(**float** a[], **int** n)

{

printf("Array:\n");

**for** (i=0;i<n;i++)

printf("%3.2f ",a[i]);

}

main()

{

**int** n, p, q;

printf("n -> "); //ввод длины массива n

scanf("%d",&n);

printf("p -> "); //ввод параметра p

scanf("%d",&p);

printf("q -> "); //ввод параметра q

scanf("%d",&q);

**float** \*a = input\_a(a, n); //функция заполнения массива

printf("\n");

//нахождение элемента с макс. косинусом

printf("Max cosinus - cos(%3.2f)\n\n",max\_cos(a, n));

sort(a, n, p, q); //сортировка массива

output\_a(a, n); //вывод массива на экран

free(a); //освобождение памяти под массив

}

**Структурная схема программы**

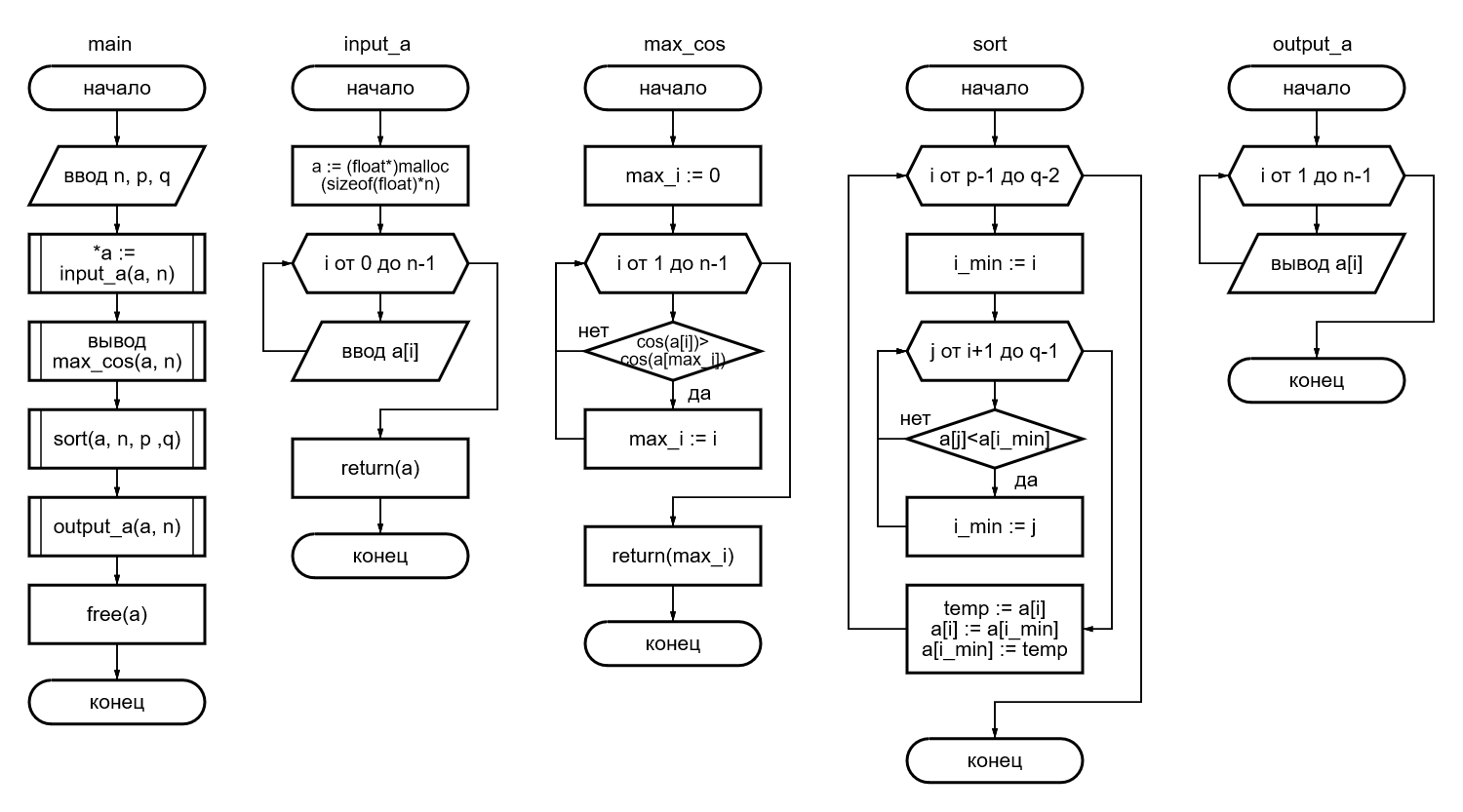


Рисунок 1 – Структурные схемы алгоритма и функций программы

**Тестовые примеры**

1. Начальные значения: n=6, p=2, q=5. Исходный массив: 6.0, 5.0, 4.0, 3.0, 2.0, 1.0. Программа должна вывести: максимальный косинус – cos(6.00), отсортированный массив – 6.00, 2.00, 3.00, 4.00, 5.00, 1.00.

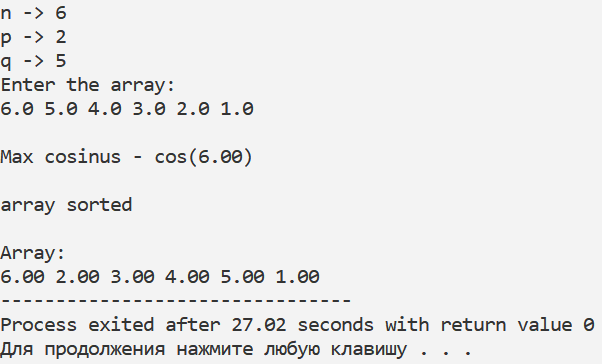


Рисунок 2 – Тестирование программы

1. Начальные значения: n=5, p=1, q=5. Исходный массив: 3.14, 1.57, 0, -1.57, -3.14. Программа должна вывести: максимальный косинус – cos(0.00), отсортированный массив – -3.14, -1.57, 0, 1.57, 3.14.

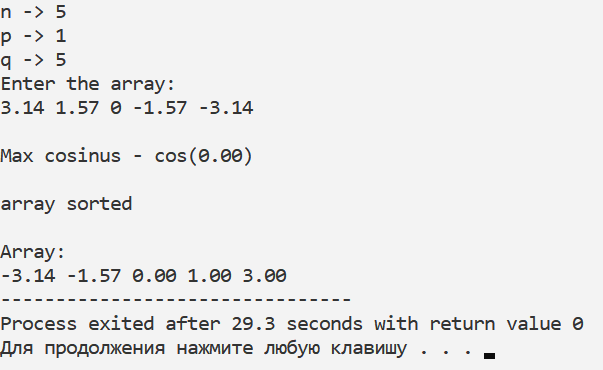


Рисунок 3 – Тестирование программы

В результате тестирования программа вывела верные результаты для всех тестовых примеров. На основании этого был сделан вывод, что она написана верно.

**Вывод**

В ходе работы были изучены особенности работы с одномерными динамическими массивами в языках C/C++, методы их обработки при помощи функций и указателей. Были получены практические навыки реализации алгоритмов создания, обработки и вывода одномерных динамических массивов средствами языков C/C++.

Результатом работы стала программа на языке C, выполняющая создание, обработку, сортировку и вывод одномерного динамического массива посредством нескольких функций в соответствии с полученным вариантом задания.