САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Практическая работа №5

«Работа с массивами»

Выполнил: Мороз И.О.

Группа К3120

Проверил: Осипов Н.А.

Санкт-Петербург

2018 г.

# Упражнение 1.

1. Для демонстрации работы массивов было создано новое приложение. С помощью цикла for был реализован ввод значений массива с клавиатуры. Также в цикле for было реализовано нахождение суммы всех элементов массива. Среднее значение находилось по формуле среднего арифметического n чисел

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

const int n = 10;

int mas[n];

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

{

cout << "mas[" << i << "] = ";

cin >> mas[i];

}

int s = 0;

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

{

s += mas[i];

}

cout << "Сумма всех элементов: " << s << "; Среднее значение: " << double(s) / n << endl;

system("pause");

return 0;

}

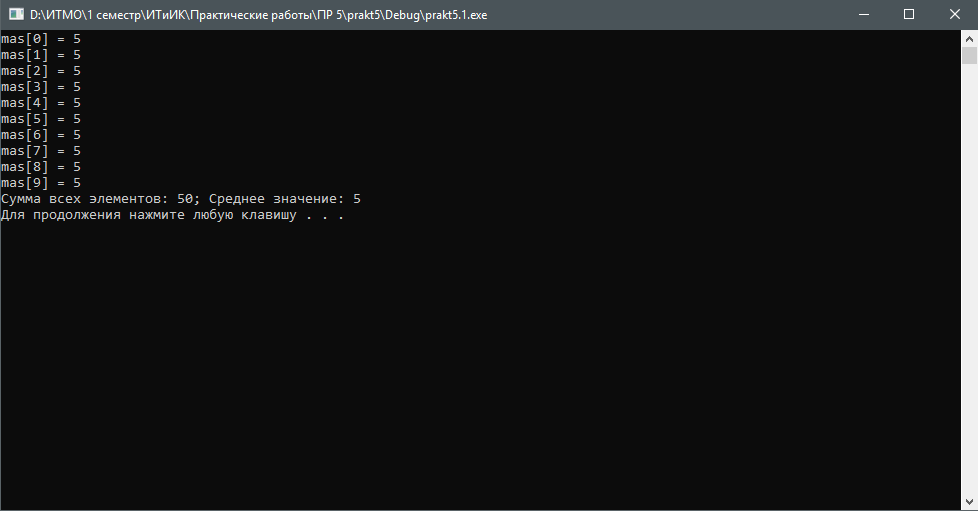


Рисунок 1 – нахождение суммы элементов массива и среднего значения

1. Было реализовано нахождение суммы положительных или отрицательных, чётных или нечётных элементов массива, нахождение минимального и максимального элементов массива, а также произведение элементов, находящихся между ними

#include <iostream>

using namespace std;

int minAB(int a, int b)

{

if (a<b)

{

return a;

}

else

{

return b;

}

}

int maxAB(int a, int b)

{

if (a > b)

{

return a;

}

else

{

return b;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

const int n = 10;

int mas[n];

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

{

cout << "mas[" << i << "] = ";

cin >> mas[i];

}

int s = 0;

int sOtr = 0;

int sPol = 0;

int sNech = 0;

int sChet = 0;

int min = mas[0];

int max = mas[0];

int minIndex = 0, maxIndex = 0;

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

{

s += mas[i];

if (mas[i] < 0)

{

sOtr += mas[i];

}

else

{

sPol += mas[i];

}

if (i % 2 != 0)

{

sNech += mas[i];

}

else

{

sChet += mas[i];

}

if (mas[i] < min)

{

min = mas[i];

minIndex = i;

}

if (mas[i] > max)

{

max = mas[i];

maxIndex = i;

}

}

int s1 = 1;

for (int i = minAB(minIndex, maxIndex) + 1; i < maxAB(minIndex, maxIndex); i++)

{

s1 \*= mas[i];

}

cout << "Сумма всех элементов: " << s << "; Среднее значение: " << double(s) / n << endl;

cout << "Сумма положительных элементов массива: " << sPol << endl;

cout << "Сумма отрицательных элементов массива: " << sOtr << endl;

cout << "Сумма чётных элементов массива: " << sChet << endl;

cout << "Сумма нечётных элементов массива: " << sNech << endl;

cout << "Минимальный элемент массива: " << min << endl;

cout << "Максимальный элемент массива: " << max << endl;

cout << "Произведение элементов между минимальным и максимальным: " << s1 << endl;

system("pause");

return 0;

}

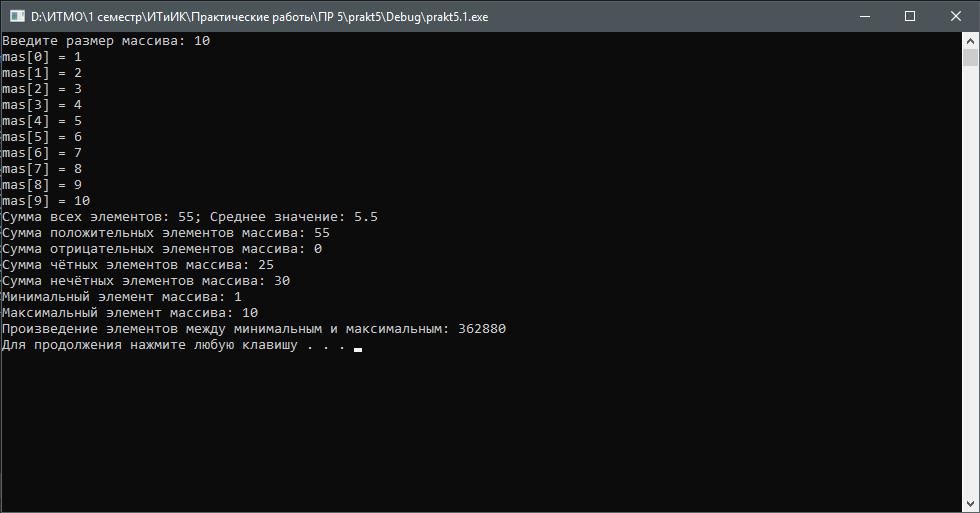


Рисунок 2 – работа усовершенствованной программы

# Упражнение 2.

1. Для демонстрации использования указателя на функцию было создано новое приложение. Оно реализует сортировку массива по возрастанию или по убыванию в зависимости от выбора пользователя. Указателем передаются функции, которые проверяют, больше одно число другого или наоборот

#include <iostream>

using namespace std;

void show\_array(const int Arr[], const int N);

bool from\_min(const int a, const int b);

bool from\_max(const int a, const int b);

void bubble\_sort(int Arr[], const int N, bool(\*compare)(int a, int b));

int main()

{

setlocale(LC\_ALL, "Russian");

const int N = 10;

int my\_choose = 0;

int A[N] = { 9,8,7,6,1,2,3,5,4,9 };

cout << "1. Сортировать по возрастанию\n";

cout << "2. Сортировать по убыванию\n";

cin >> my\_choose;

cout << "Исходные данные: ";

show\_array(A, N);

switch (my\_choose)

{

case 1: bubble\_sort(A, N, from\_min); break;

case 2: bubble\_sort(A, N, from\_max); break;

default: cout << "\rНеизвестная операция ";

}

show\_array(A, N);

system("pause");

return 0;

}

void show\_array(const int Arr[], const int N)

{

for (int i = 0; i < N; i++)

cout << Arr[i] << " ";

cout << "\n";

}

bool from\_min(const int a, const int b)

{

return a > b;

}

bool from\_max(const int a, const int b)

{

return a < b;

}

void bubble\_sort(int Arr[], const int N, bool(\*compare)(int a, int b))

{

for (int i = 1; i < N; i++)

{

for (int j = 0; j < N - 1; j++)

{

if ((\*compare)(Arr[j], Arr[j + 1])) swap(Arr[j], Arr[j + 1]);

}

}

}

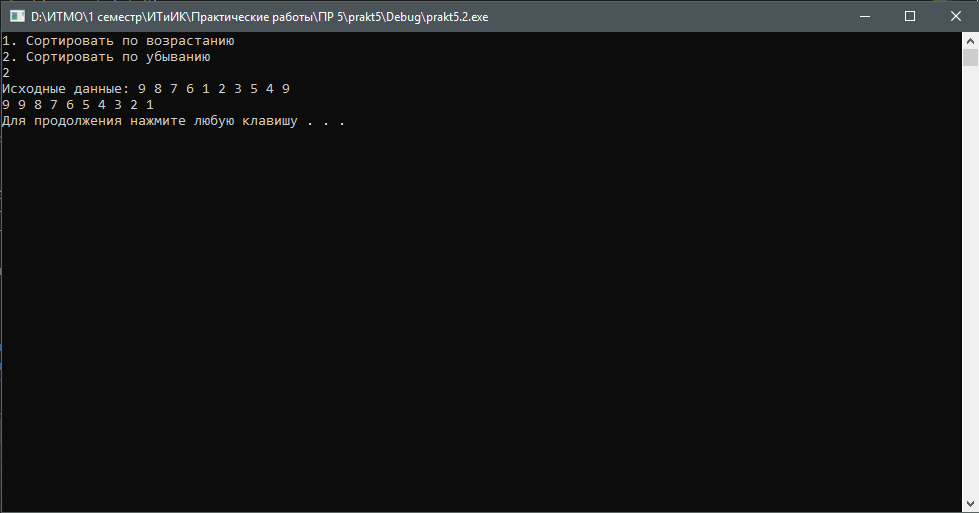


Рисунок 3 – сортировка массива по убыванию

1. В следующей части упражнения был реализован массив указателей на функции. При использовании такого массива можно отказаться от конструкции switch-case

int main()

{

setlocale(LC\_ALL, "Russian");

bool(\*from\_f[2])(int, int) = { from\_min,from\_max };

const int N = 10;

int my\_choose = 0;

int A[N] = { 9,8,7,6,1,2,3,5,4,9 };

cout << "1. Сортировать по возрастанию\n";

cout << "2. Сортировать по убыванию\n";

cin >> my\_choose;

cout << "Исходные данные: ";

show\_array(A, N);

bubble\_sort(A, N, (\*from\_f[my\_choose - 1]));

show\_array(A, N);

system("pause");

return 0;

}

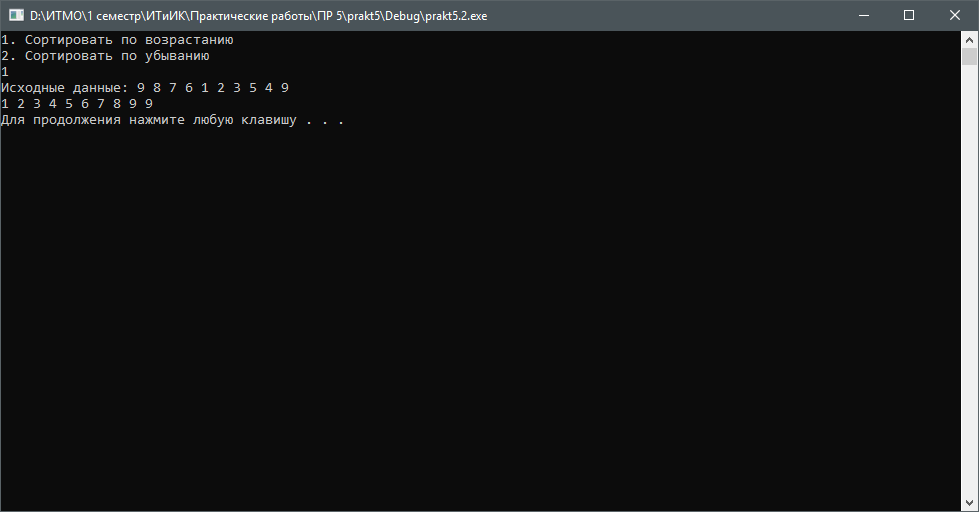


Рисунок 4 – изменения не повлияли на работоспособность алгоритма

# Упражнение 3.

1. Предыдущие программы были изменены так, чтобы испольовались динамические массивы. Пример реализации динамического массива из первой программы

cout << "Введите размер массива: ";

int n = 0;

cin >> n;

int\* mas = new int[n];

После работы программы необходимо очистить используемую динамическую память

delete[] mas;

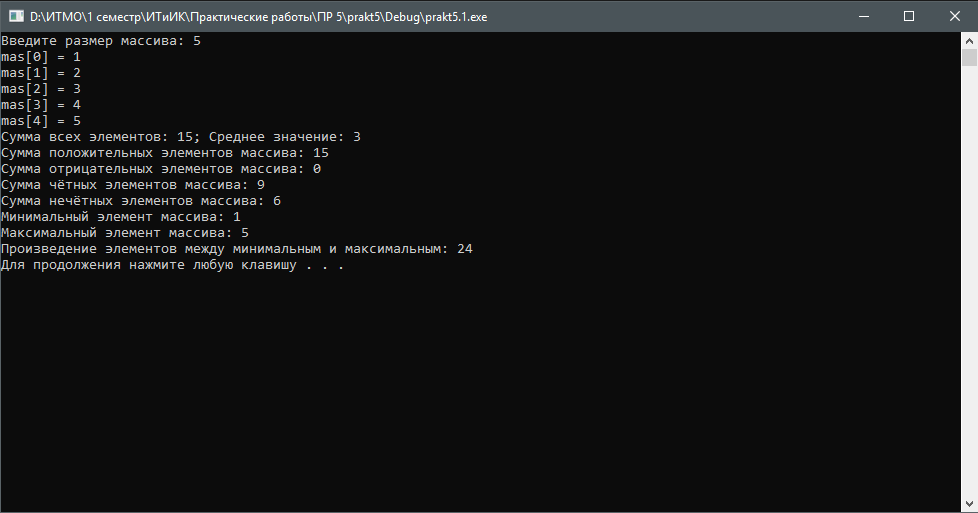


Рисунок 5 – появилась возможность изменять размеры массива

1. В программе из упражнения 2, кроме ввода размера массива, был реализован также ввод самого массива с клавиатуры

cout << "Введите размер массива: ";

int N = 0;

cin >> N;

int\* A = new int[N];

for (int i = 0; i < N; i++)

{

cout << "A[" << i << "] = ";

cin >> A[i];

}

//…

delete[] A;

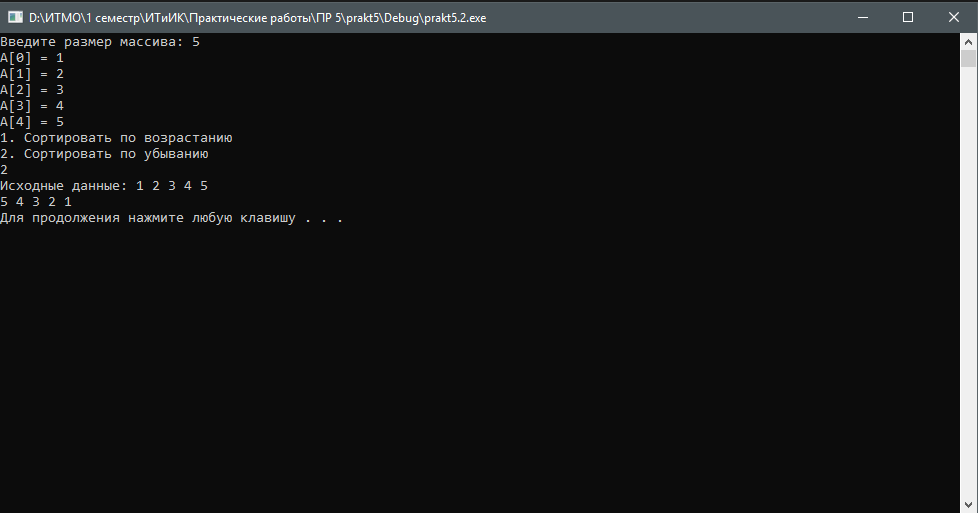


Рисунок 6 – сортировка отлично работает с динамическим массивом

# Вывод

Во время выполнения данной работы были получены навыки работы с массивами. Рассмотренны указатели на функции. Были использованы возможности динамических массивов.

# Контрольные вопросы

1. Что будет, если передать в массив больше 10 значений?

Ответ: произойдёт переполнение массива, программа выдаст ошибку.

1. За счёт чего появилась возможность отказаться от switch-case?

Ответ: был заменён оператор выбора на массив из указателей на функции. Выбор производился при помощи обращения к элементу массива.