# Лабораторная работа 5

**Сортировка одномерного числового массива.**

**Цели работы:**

- освоение методов обработки массивов на примере сортировки массива;

- знакомство с алгоритмами сортировки;

- использование динамических массивов;

-инициализация массивов с помощью датчика случайных чисел;

- оценка быстродействия алгоритма.

**Задание:**

Отсортировать числовой массив методом выбора максимального (минимального) элемента и методом пузырькового всплытия. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов.

Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного не отсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.

Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализуйте с помощью динамического массива, а для его инициализации используйте датчик случайных чисел (см. Приложение 1). Результаты исследования выведите в виде отформатированной таблицы.

При выполнении работы обратите внимание на следующие требования и рекомендации:

1. Размерность нединамического массива может быть только константой или константным выражением. Рекомендуется задавать размерность с помощью именованной константы.
2. Элементы массивов нумеруются с нуля, поэтому максимальный номер элемента всегда на единицу меньше размерности.
3. Автоматический контроль выхода индекса за границы массива не производится, поэтому программист должен следить за этим самостоятельно.
4. Указатель — это переменная, в которой хранится адрес области памяти.
5. Имя массива является указателем на его нулевой элемент.
6. Обнуления динамической памяти при ее выделении не происходит.
7. Освобождение памяти, выделенной посредством new[], выполняется с помощью  
   операции delete[].
8. Перед выходом локального указателя из области его действия следует освобождать связанную с ним динамическую память.
9. Если количество элементов, которые должны быть введены в программу, известно до ее выполнения, определяйте массив в операторе описания переменных  
   (причем лучше как локальную переменную, чем как глобальную);

если количество можно задать во время выполнения программы, но до ввода элементов, создавайте динамический массив.

1. Алгоритмы сортировки массивов различаются по быстродействию и занимаемой памяти, причем эти характеристики зависят от упорядоченности сортируемого массива.

**Приложение 1**

**Использование датчика случайных чисел**.

#include <time.h> // time

#include <stdlib.h> // srand, rand

#include <iostream>

using namespace std;

int iRandom( int a, int b ) // целое из [a,b]

{

return a + ( b - a +1) \* rand()/RAND\_MAX ;

}

double dRandom( int a, int b ) // вещественное из [a,b]

{

return a + ( b - a ) \* rand() / (double)RAND\_MAX;

}

void main()

{

const int n = 13; // количество элементов в массиве

int a[n], i; // массив целых чисел

srand( (unsigned int) time( NULL ) ); // начальное число-время

rand( ); //сброс первого числа, чтобы не

// повторялось, пока srand не изменяется

for( i = 0; i < n; i++ ) // цикл инициализации массива

{ a[i] = iRandom( 1, 9 ); // целая случайная величина

cout<<”a[“<<i<<”]=”<<a[i]<<endl;

}

}