Цель работы: Получение практических навыков при работе с массивами.

1. Постановка задачи

1. Сформировать массив из n элементов с помощью датчика случайных чисел (n задается пользователем с клавиатуры)
2. Распечатать полученный массив
3. Выполнить удаление минимального элемента из массива
4. Вывести полученный результат
5. Выполнить добавление элемента с номером K в массив
6. Вывести полученный результат
7. Выполнить перестановку минимального и максимального элементов в массиве
8. Вывести полученный результат
9. Выполнить поиск элемента, равного среднему арифметическому элементов массива, в массиве элементов и посчитать количество сравнений, необходимых для поиска нужного элемента
10. Вывести полученный результат
11. Выполнить сортировку простым выбором массива указанным методом
12. Вывести полученный результат
13. Выполнить поиск указанных элементов, равного среднему арифметическому элементов массива, в отсортированном массиве и посчитать количество сравнений, необходимых для поиска нужного элемента
14. Вывести полученный результат

2.

Используем пространство имен std для работы со строками. Инициализируем массив, с заранее заданным количеством элементов. Инициализируем функцию для простой сортировки. Принцип простой сортировки:

Идем от i=1 до size-1, где size-количество элементов в массиве, элемента массива

Записываем в переменную min номер минимального элемента

Запускаем цикл от i+1 до size и ищем номер элемента, который будет меньше элемента с номер min

Делаем перестановку этих двух элементов

Инициализируем функцию main и все необходимые переменные для работы программы.

Введем n, при этом n меньше размера массива и больше нуля.

Заполним массив случайными числами от 1 до 500 и выведем

Найдем номер минимального элемента через цикл, присвоим 0 этому элементу. Сделаем сдвиг элементов. Выведем полученный массив

Введем номер элемента, который нужно добавить. Выполним сдвиг элементов так, чтобы элемент с указанным номером оказался пустым. Присвоим ранее указанному элементу случайное значение от 1 до 500. Выведем полученный массив.

Найдем минимальное и максимальное число и их номера. Сделаем перестановку и выведем полученный массив.

Найдем среднее арифметическое путем сложения всех элементов через цикл, поделим полученное число на количество элементов. Если мы нашли нужный нам элемент, то выходим из цикла и выводим успех, добавляя количество сравнений. Если такой элемент не нашелся, то выводим неудачу и количество сравнений, выполненных в процессе работы алгоритма.

Выполним сортировку ранее озвученным алгоритмом простой сортировки. Выведем полученный массив.

Найдем среднее арифметическое путем сложения всех элементов через цикл, поделим полученное число на количество элементов. Если мы нашли нужный нам элемент, то выходим из цикла и выводим успех, добавляя количество сравнений. Если такой элемент не нашелся, то выводим неудачу и количество сравнений, выполненных в процессе работы алгоритма.

Завершим работу программы.

3.

4.

#include <iostream>  
  
using namespace std;  
  
// Размер массива  
#define ARRAY\_SIZE 100  
  
const int MAX\_INT = 500; // Максимальный элемент  
const int MIN\_INT = 1; // Минимальный элемент  
  
// Количество элементов  
int n;  
  
// Массив  
int a[ARRAY\_SIZE];  
  
// Сортировка простым выбором  
void simpleSort(int \*num, int size) {  
 int min = 0, temp = 0;  
 for (int i = 0; i < size-1; i++) {  
 min = i;  
 for (int j = i+1; j < size; j++)  
 {  
 if (num[j] < num[min])  
 min = j;  
 }  
 temp = num[i];  
 num[i] = num[min];  
 num[min] = temp;  
 }  
}  
  
int main() {  
 int min = MAX\_INT+1; // Минимальное число  
 int minI = -1; // i минимального числа  
 int max = MIN\_INT-1; // Максимальное число  
 int maxI = -1; // i максимального числа  
 int k = 0; // Временная переменная  
  
 cout << "Enter: N" << endl;  
 do {  
 cin >> n;  
 } while (n >= ARRAY\_SIZE || !n);  
 cout << endl;  
  
 // Заполнили и вывели (1, 2)  
 for(int i = 0; i < n; i++) {  
 a[i] = rand()%MAX\_INT+MIN\_INT;  
 cout << i << ") " << a[i] << endl;  
 }  
 cout << endl;  
  
 // Удалить (3, 4)  
 for(int i = 0; i < n; i++) {  
 if (min > a[i]) {  
 min = a[i];  
 minI = i;  
 }  
 }  
 a[minI] = 0;  
 for(int i = minI; i < n; i++) {  
 a[i] = a[i+1];  
 }  
  
 for(int i = 0; i < n; i++) {  
 cout << i << ") " << a[i] << endl;  
 }  
 cout << endl;  
  
 // Добавление (5, 6)  
 cout << "Enter: K" << endl;  
 do {  
 cin >> k;  
 } while (k < 0 || k >= n);  
  
 for(int i = n-1; i >= k; i--) {  
 a[i] = a[i-1];  
 }  
 a[k] = rand()%MAX\_INT+MIN\_INT;  
 for(int i = 0; i < n; i++) {  
 cout << i << ") " << a[i] << endl;  
 }  
 cout << endl;  
 // Перестановка (7, 8)  
  
 // Обнуляем  
 min = MAX\_INT+1; // Минимальное число  
 minI = -1; // i минимального числа  
  
 for(int i = 0; i < n; i++) {  
 if(a[i] > max) {  
 max = a[i];  
 maxI = i;  
 } else if(a[i] < min) {  
 min = a[i];  
 minI = i;  
 }  
 }  
 k = a[maxI]; // Запомнили сколько было  
 a[maxI] = a[minI]; // Поменяли максимум на минимум  
 a[minI] = k; // Поменяли минимум на максимум  
 for(int i = 0; i < n; i++) {  
 cout << i << ") " << a[i] << endl;  
 }  
 cout << endl;  
  
 // Поиск указанных в массиве элементов  
 // Среднее арифметическое (9, 10)  
 int average = 0;  
 int counter = 0;  
 for(int i = 0; i < n; i++) {  
 average += a[i];  
 }  
 k = -1;  
 average /= n;  
 for(int i = 0; i < n; i++) {  
 if (average == a[i]) {  
 k = i;  
 break;  
 }  
 else  
 counter++;  
 }  
 if(k >= 0)  
 cout << "Success: Checks(" << counter << "), element " << k << endl;  
 else  
 cout << "Fail: Checks(" << counter << ")" << endl;  
 cout << endl;  
  
 // Сортировка массива (11, 12)  
 simpleSort(a, n);  
 for(int i = 0; i < n; i++) {  
 cout << i << ") " << a[i] << endl;  
 }  
 cout << endl;  
  
 // Поиск (13, 14)  
 counter = 0, k = -1;  
 for(int i = 0; i < n; i++) {  
 if (average == a[i]) {  
 k = i;  
 break;  
 }  
 else  
 counter++;  
 }  
 if(k >= 0)  
 cout << "Success: Checks(" << counter << "), element " << k << endl;  
 else  
 cout << "Fail: Checks(" << counter << ")" << endl;  
  
 return 0;  
}

5.

