Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа №3

«Работа с одномерными массивами»

Вариант 12

Выполнил студент группы ИВТ-20-1б

Копыгин Игорь Евгеньевич

Проверил старший преподаватель

Ярулин Денис Владимирович

Пермь 2020

**Цель работы:**

**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 указанных элементов из массива.

4) Вывести полученный результат.

5) Выполнить добавление указанных элементов в массив.

6) Вывести полученный результат.

7) Выполнить перестановку элементов в массиве.

8) Вывести полученный результат.

9) Выполнить поиск указанных в массиве элементов и подсчитать количество сравнений, необходимых для поиска нужного элемента.

10) Вывести полученный результат.

11) Выполнить сортировку массива указанным методом.

12) Вывести полученный результат.

13) Выполнить поиск указанных элементов в отсортированном массиве и подсчитать количество сравнений, необходимых для поиска нужного элемента.

14) Вывести полученный результат.

**Вариант:**

Удаление: элемент с заданным номером.

Добавление: элемент с номером К.

Перестановка: четные элементы переставить в начало массива, нечетные - в конец.

Поиск: элемент равный среднему арифметическому элементов массива.

Сортировка: простое включение.

**Анализ задачи:**

Объявим статичный массив размером 100. Предположим, что массивы большего размера рассматриваться не будут. Прочитаем ввод пользователя в переменную n - это размер нашего массива. С помощью цикла for запишем в элементы массива с [0] по [n-1] псевдорандомные числа от 0 до 10, предварительно задав seed для функции rand() с помощью функции srand(). Выведем элементы данного массива с помощью цикла for, на каждой итерации которого будем выводить очередной элемент массива, разделяя их пробелом.

Прочитаем ввод пользователя в переменную m - это номер элемента, который необходимо удалить. Для удаления элемента из массива с номером m будем ипользовать цикл for от m до n невключительно. На каждом шагу цикла будем менять текущий элемент на следущий. Поскольку наш массив уменьшился на 1, вычитаем единицу из n. Выведем получившийся массив.

Читаем ввод пользователя в k и elem - номер элемента, который нужно вставить, и сам элемент соответственно. Поскольку мы увеличиваем массив, увеличим n на 1. Для вставки элемента в миссив будем ипользовать цикл for от k до n невключительно. На каждом шаге цикла будем обменивать значения в elem и текущем элементе массива. Для этого объявим перемнную tmp и запишем в неё текущий элемент, заменим текущий элемет массива на значение elem, elem поменяем на значение tmp. Выведем получившийся массив.

Для перестановки четных и нечетный элементов пройдемся по всему массиву циклом for. На каждом шагу будем проверять элемент на нечетность. Если очередной элемент нечетный, то найдем самый последний четный элемент, стоящий после текущего нечетного. Если такой элемент не найден, то в нашем массиве уже все четные в начале, а нечетные в конце и можно завершать данный цикл. Если элемент найден, то поменяем его с рассматриваемым нечетным. Выведем получившийся массив.

Объявим три переменные типа int: mean = 0, nmean = -1, ncompare = 0 - среднее арифметическое элеементов массива, номер элемента, равного среднему арифметическому, и число сравнений соответсвенно. Пройдемся по массиву циклом for и сложим все элементы массива в переменной mean. Разделим mean на n. Поскольку mean типа int, дробная часть от деления будет отброшена. Использовать для этого float не иммет особого смысла, так как такой элмент почти никогда не найдется в массиве. Пройдемся по массиву циклом for, на каждом шагу увеличивая число сравнений на один и проверяя текущий элемент на равенство со средним значением. После выхода из цикла в случае нахождения такого элемента выведем среднее значение, номер этого элемента и число сравнений, которые потребовались, иначе скажем, что такого элемента нет.

Для сортировки простым включением пройдемся по массиву циклом for. На шаге цикла создадим перемнные j и elem, в которые запишем номер текущего элеменнта - 1 и значение текущего элемента соответственно. С помощью цикла while пройдемся от j-ого элмента до тех пор, пока j неотрицательное и j-ый элемент больше значения, сохраненного в elem. На каждом шаге цикла будем присваивать следущему после j-ого элемента значение j-ого элемента, после чего будем уменьшать j на 1. После выхода из цикла while присвоим следущему после j-ого элементу значение переменной elem. После выхода из цикла while выведем отсортированный массив.

Заново найдем средний элемент и выведем полученный результат.

**Код программы:**

#include <iostream>

#include <ctime>

**using** **namespace** std;

**int** **main**()

{

**int** a[**100**];

**int** n, m;

srand(time(**0**));

cout << "Size of array: ";

cin >> n;

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

{

a[i] = rand() % **10**;

}

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

{

cout << a[i] << ' ';

}

cout << endl << "Element number to delete: ";

cin >> m;

**for** (**int** i = m; i < n; i++)

{

a[i] = a[i + **1**];

}

n--;

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

{

cout << a[i] << ' ';

}

cout << endl << "Element number to insert: ";

**int** k, elem;

cin >> k;

cout << "Element: ";

cin >> elem;

n++;

**for** (**int** i = k; i < n; i++)

{

**int** tmp = a[i];

a[i] = elem;

elem = tmp;

}

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

{

cout << a[i] << ' ';

}

cout << endl;

**int** even;

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

{

**if** (a[i] % **2** == **1**)

{

even = -**1**;

**for** (**int** j = i + **1**; j < n; j++)

{

**if** (a[j] % **2** == **0**)

{

even = j;

}

}

**if** (even == -**1**)

{

**break**;

}

**int** tmp = a[i];

a[i] = a[even];

a[even] = tmp;

}

}

cout << "Array after putting even numbers at the beginning and odd at the end: ";

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

{

cout << a[i] << ' ';

}

**int** mean = **0**, nmean = -**1**, ncompare = **0**;

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

{

mean += a[i];

}

mean /= n;

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

{

ncompare++;

**if** (a[i] == mean)

{

nmean = i;

**break**;

}

}

cout << endl;

**if** (nmean == -**1**)

{

cout << "No mean element found";

}

**else**

{

cout << "Mean is " << mean << endl;

cout << "Mean element number is " << nmean << endl;

cout << "It took " << ncompare << " comparisons to find it";

}

cout << endl;

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

{

**int** j = i - **1**, elem = a[i];

**while** (j >= **0** && elem < a[j])

{

a[j + **1**] = a[j];

j--;

}

a[j + **1**] = elem;

}

cout << "Array after sorting: ";

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

{

cout << a[i] << ' ';

}

cout << endl;

ncompare = **0**;

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

{

ncompare++;

**if** (a[i] == mean)

{

nmean = i;

**break**;

}

}

**if** (nmean == -**1**)

{

cout << "No mean element found";

}

**else**

{

cout << "Mean is " << mean << endl;

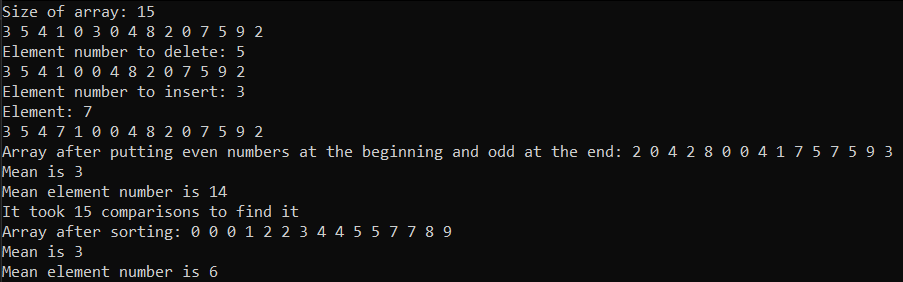
cout << "Mean element number is " << nmean << endl;

cout << "It took " << ncompare << " comparisons to find it";

}

}

**Результат работы программы:**

****