Министерство цифрового развития, связи и массовых коммуникаций РФ

Федеральное государственное бюджетное общеобразовательное учреждение

высшего образования

“Сибирский Государственный Университет Телекоммуникаций и Информатики”

(СибГУТИ)

Кафедра телекоммуникационных систем и вычислительных средств

Отчет по лабораторной работе 2 на тему: Указатели и динамические массивы. Использование указателей в качестве аргументов функций

Выполнил: студент 1 курса группы ИА-231

Зырянов Иван Александрович

Проверила: Моренкова Ольга Ильинична

Новосибирск, 2023

# Задание:

Вариант 9:

В целочисленном массиве Х(N) удалить все элементы, расположенные между максимальным и минимальным элементами. Найти среднее арифметическое элементов массива до и после удаления. В программе написать функции: удаления элемента, поиска номеров минимального и максимального элементов, определения среднего арифметического элементов массива.

Схема алгоритма решения задачи:

1. Инициализировать массив X(N) размером N.
2. Определить функцию **find\_min\_max\_indexes(X, N)**, которая принимает массив X и его размер N и возвращает номера минимального и максимального элементов в виде кортежа (min\_index, max\_index).
   * Инициализировать переменные min\_index и max\_index со значением 0.
   * Инициализировать переменные min\_value и max\_value со значением X[0].
   * Для каждого индекса i от 1 до N-1:
     + Если X[i] < min\_value, обновить min\_value и min\_index.
     + Если X[i] > max\_value, обновить max\_value и max\_index.
   * Вернуть кортеж (min\_index, max\_index).
3. Определить функцию **delete\_elements(X, start\_index, end\_index)**, которая принимает массив X, начальный индекс start\_index и конечный индекс end\_index, и удаляет элементы массива между указанными индексами.
   * Создать новый пустой массив Z.
   * Добавить в массив Z все элементы X с индексами от 0 до start\_index.
   * Добавить в массив Z все элементы X с индексами от end\_index до N-1.
   * Вернуть массив Z.
4. Определить функцию **calculate\_mean(X, N)**, которая принимает массив X и его размер N и вычисляет среднее арифметическое его элементов.
   * Инициализировать переменную sum\_value со значением 0.
   * Для каждого элемента elem в массиве X:
     + Увеличить sum\_value на elem.
   * Вычислить среднее арифметическое как sum\_value / N.
   * Вернуть среднее арифметическое.
5. Вызвать функцию **find\_min\_max\_indexes(X, N)** и сохранить результат в переменных min\_index и max\_index.
6. Вызвать функцию **delete\_elements(X, min\_index, max\_index)** и сохранить результат в массиве Z.
7. Вызвать функцию **calculate\_mean(X, N)** и сохранить результат в переменной mean\_X.
8. Вызвать функцию **calculate\_mean(Z, new\_N)** (где new\_N - новый размер массива Z) и сохранить результат в переменной mean\_Z.
9. Вывести массивы X и Z, а также значения mean\_X и mean\_Z.

Код:

#include <iostream>

#include <algorithm>

// Функция удаления элемента из массива

void removeElement(int arr[], int& size, int index) {

    if (index < 0 || index >= size) {

        std::cout << "Неверный индекс элемента для удаления." << std::endl;

        return;

    }

    for (int i = index; i < size - 1; i++) {

        arr[i] = arr[i + 1];

    }

    size--;

}

// Функция поиска индекса минимального элемента в массиве

int findMinIndex(const int arr[], int size) {

    int minIndex = 0;

    for (int i = 1; i < size; i++) {

        if (arr[i] < arr[minIndex]) {

            minIndex = i;

        }

    }

    return minIndex;

}

// Функция поиска индекса максимального элемента в массиве

int findMaxIndex(const int arr[], int size) {

    int maxIndex = 0;

    for (int i = 1; i < size; i++) {

        if (arr[i] > arr[maxIndex]) {

            maxIndex = i;

        }

    }

    return maxIndex;

}

// Функция вычисления среднего арифметического элементов массива

double calculateAverage(const int arr[], int size) {

    if (size == 0) {

        return 0.0;

    }

    int sum = 0;

    for (int i = 0; i < size; i++) {

        sum += arr[i];

    }

    return static\_cast<double>(sum) / size;

}

int main() {

    int size;

    std::cout << "Enter the size of the array: ";

    std::cin >> size;

    int\* X = new int[size];

    std::cout << "Enter array elements:" << std::endl;

    for (int i = 0; i < size; i++) {

        std::cout << "X[" << i << "]: ";

        std::cin >> X[i];

    }

    int minIndex = findMinIndex(X, size);

    int maxIndex = findMaxIndex(X, size);

    if (minIndex > maxIndex) {

        std::swap(minIndex, maxIndex);

    }

    // Удаление элементов между минимальным и максимальным

    int elementsToRemove = maxIndex - minIndex - 1;

    for (int i = 0; i < elementsToRemove; i++) {

        removeElement(X, size, minIndex + 1);

    }

    // Вычисление среднего арифметического элементов до и после удаления

    double averageBefore = calculateAverage(X, size);

    double averageAfter = calculateAverage(X, size - elementsToRemove);

    std::cout << "Arithmetic mean of elements before removal: " << averageBefore << std::endl;

    std::cout << "Arithmetic mean of elements after deletion: " << averageAfter << std::endl;

    delete[] X;

    return 0;

}

Вывод программы:

