

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 6

з дисципліни «Алгоритмізація та програмування»

Тема: *«Реалізація алгоритмів обробки одновимірних
масивів мовою C ++»*

ХАІ.301. спец . 319 . номер 6

Виконав студент гр. 319

Шаньгін Андрій

(підпис, дата)

(П.І.Б.)

Перевірив

_____ к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата)

(П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення одновимірних і масивів на мові C ++ і реалізувати декларацію, введення з консолі, обробку і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1(36). Вирішити завдання на аналіз і виведення елементів одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести весь масив у рядок в порядку зростання індексів, потім – елементи чи підраховані результати відповідно до завдання. Дан масив розміру N. Знайти максимальний з його елементів, які не є ні локальним мінімумом, ні локальним максимумом (визначення локального мінімуму і локального максимуму дані в завданнях Array32 Дан масив розміру N. Знайти номер його першого локального мінімуму (локальний мінімум - це елемент, який менше будь-якого зі своїх сусідів). і Array33 Дан масив розміру N. Знайти номер його останнього локального максимуму (локальний максимум - це елемент, який більше будь-якого зі своїх сусідів).). Якщо таких елементів в масиві немає, то вивести 0 (як дійсне число).

Завдання 2. Вирішити завдання на перетворення одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести у консоль заданий масив, потім – змінений. Дан масив розміру N, всі елементи якого, крім першого, впорядковані за зростанням. Зробити масив упорядкованим, перемістивши перший елемент на нову позицію.

Завдання 3. У функції main() організувати багаторазовий вибір одного з двох завдань. Кожне завдання описати окремою функцією без параметрів. Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі розділ 1 і номер задачі 36

Вхідні дані (ім'я, опис, тип, обмеження):

Дан масив розміру N. Знайти максимальний з його елементів, які не є ні локальним мінімумом, ні локальним максимумом (визначення локального мінімуму і локального максимуму дані в завданнях Array32 Дан масив розміру N. Знайти номер його першого локального мінімуму (локальний мінімум - це елемент, який менше будь-якого зі своїх сусідів). і Array33 Дан масив розміру N. Знайти номер його останнього локального максимуму (локальний максимум - це елемент, який більше будь-якого зі своїх сусідів).). Якщо таких елементів в масиві немає, то вивести 0 (як дійсне число).

Алгоритм вирішення показано на рис. 1

```

Меню:
1. Завдання 36
2. Завдання 87
3. Вийти з програми
Виберіть пункт меню: 1
Введіть розмір масиву N: 2
Масив повинен мати хоча б 3 елементи.
  
```

Рисунок 1 – <компіляція завдання 1>

Лістинг коду вирішення задачі розділ 1 і номер задачі 36 наведено в дод. А

Завдання 2.

Вирішити завдання на перетворення одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести у консоль заданий масив, потім – змінений. Дан масив розміру N, всі елементи якого, крім першого, впорядковані за зростанням. Зробити масив упорядкованим, перемістивши перший елемент на нову позицію.

Вирішення задачі розділ 2 і номер задачі 87

Алгоритм вирішення показано на рис. 2

```

меню:
1. Завдання 36
2. Завдання 87
3. Вийти з програми
Виберіть пункт меню: 2
Введіть розмір масиву N: 3 4 5 6 7 8
Введіть елементи масиву: Array87: Упорядкований масив: 4 5 6

```

Рисунок 2 – <компіляція завдання 2>

Лістинг коду вирішення задачі розділ 2 і номер задачі 87 наведено в дод. А

Завдання 3.

Вирішення задачі розділ 3

Вхідні дані (ім'я, опис, тип, обмеження):

У функції main() організувати багаторазовий вибір одного з двох завдань. Кожне завдання описати окремою функцією без параметрів. Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами.

Алгоритм вирішення показано на рис. 3

```

Меню:
1. Завдання 36
2. Завдання 87
3. Вийти з програми

```

Рисунок 3 – <компіляція завдання 3>

Лістинг коду вирішення задачі розділ 3 наведено в дод. А

ВИСНОВКИ

У процесі виконання роботи на тему "Реалізація алгоритмів обробки одновимірних масивів мовою C++" було вивчено основні теоретичні відомості щодо представлення одновимірних масивів у мові програмування C++. Зокрема, розглянуто принципи оголошення масивів, їх ініціалізації, заповнення даними за допомогою введення з клавіатури, а також методи обробки та виведення результатів у консоль. Практично було реалізовано алгоритми для введення, обробки та виведення одновимірних масивів у середовищі Visual Studio.

Отримані навички дозволяють ефективно працювати з масивами, виконувати різноманітні обчислення (пошук максимального або мінімального елемента, перетворення одновимірного масиву), а також виводити дані у зручному вигляді для користувача. Таким чином, під час роботи були здобуті важливі теоретичні знання та практичні навички, необхідні для подальшого вивчення алгоритмізації та програмування мовою C++.

ДОДАТОК А

Лістинг коду програми до задач

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <climits> // Для INT_MIN
using namespace std;

// Функція для перевірки, чи є елемент локальним мінімумом або максимумом
bool isLocalExtremum(const vector<int>& arr, int index) {
    if (index <= 0 || index >= arr.size() - 1) return false;
    return (arr[index] > arr[index - 1] && arr[index] > arr[index + 1]) ||
           (arr[index] < arr[index - 1] && arr[index] < arr[index + 1]);
}

void solveArray36(const vector<int>& arr) {
    int maxElement = INT_MIN;
    bool found = false;

    // Пошук максимального елемента, який не є локальним мінімумом/максимумом
    for (int i = 0; i < arr.size(); i++) {
        if (!isLocalExtremum(arr, i)) {
            if (arr[i] > maxElement) {
                maxElement = arr[i];
                found = true;
            }
        }
    }

    // Виведення результату
    if (found) {
        cout << "Array36: Максимальний елемент, що не є локальним екстремумом: "
<< maxElement << endl;
    } else {
        cout << "Array36: Таких елементів немає. Результат: 0" << endl;
    }
}

void solveArray87(vector<int>& arr) {
    int firstElement = arr[0];
    arr.erase(arr.begin()); // Видаляємо перший елемент
    sort(arr.begin(), arr.end()); // Сортуємо решту масиву
    arr.insert(upper_bound(arr.begin(), arr.end(), firstElement), firstElement);
    // Вставляємо перший елемент у потрібну позицію

    // Виведення результату

```

```

    cout << "Array87: Упорядкований масив: ";
    for (int num : arr) {
        cout << num << " ";
    }
    cout << endl;
}

void findMinMaxIndices(const vector<int>& arr) {
    int minIndex = 0, maxIndex = 0;

    for (int i = 1; i < arr.size(); i++) {
        if (arr[i] < arr[minIndex]) minIndex = i;
        if (arr[i] > arr[maxIndex]) maxIndex = i;
    }

    cout << "Індекс мінімуму: " << minIndex << ", значення: " << arr[minIndex] <<
endl;
    cout << "Індекс максимуму: " << maxIndex << ", значення: " << arr[maxIndex]
<< endl;
}

int main() {
    int N;
    int choice;

    do {
        // Меню вибору
        cout << "Меню:\n";
        cout << "1. Завдання 36\n";
        cout << "2. Завдання 87\n";
        cout << "3. Вийти з програми\n";
        cout << "Виберіть пункт меню: ";
        cin >> choice;

        if (choice == 1 || choice == 2) {
            cout << "Введіть розмір масиву N: ";
            cin >> N;

            if (N < 3) {
                cout << "Масив повинен мати хоча б 3 елементи." << endl;
                continue;
            }

            vector<int> arr(N);
            cout << "Введіть елементи масиву: ";
            for (int i = 0; i < N; i++) {
                cin >> arr[i];
            }

            // Виконання обраного завдання

```

```
        if (choice == 1) {
            solveArray36(arr);
            findMinMaxIndices(arr);
        } else if (choice == 2) {
            solveArray87(arr);
        }
    } else if (choice != 3) {
        cout << "Невірний вибір, спробуйте ще раз.\n";
    }

} while (choice != 3);

cout << "Програма завершена.\n";
return 0;
}
```


ДОДАТОК Б
Скрін-шоти вікна виконання програми

рисунок

Рисунок Б.1 – Екран виконання програми для вирішення завдання
назва та номер

рисунок

Рисунок Б.2 – Екран виконання програми для вирішення завдання
назва та номер