

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №5

за курсом “Програмування”

студентки групи ПА-23-1

Мазур Віолети

кафедра комп’ютерних технологій, ДНУ

2023/2024

1. Постановка задачі

Індивідуальне завдання. Варіант 12

Задане натуральне число n , цілі числа a_1, \dots, a_n . Залишити без змін послідовність a_1, \dots, a_n , якщо її члени упорядковані за зростанням або за спаданням. В протилежному випадку отримати послідовність a_1, \dots, a_m ($m < n$), ще m є таким, що або $a_1 \leq a_2 \leq \dots \leq a_m$ та $a_m > a_{m+1}$, або $a_1 \geq a_2 \geq \dots \geq a_m$ та $a_m < a_{m+1}$.

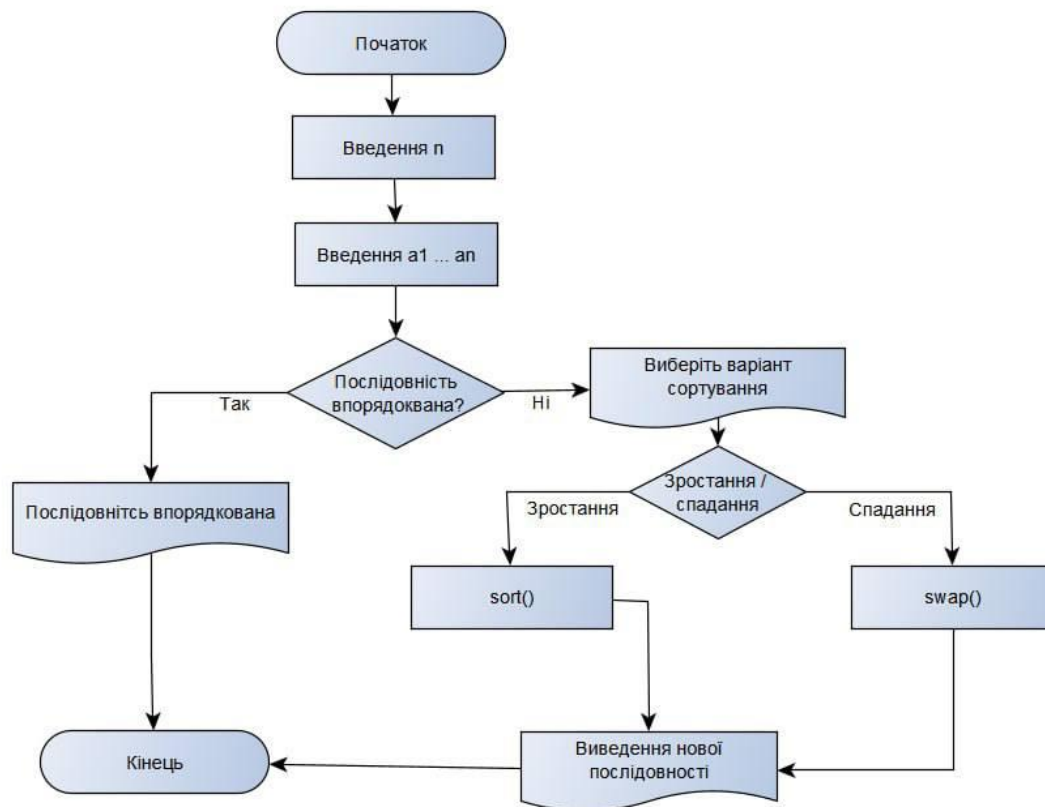
Не використовувати масиви.

2. Опис розв'язку + блок-схема

Опис розв'язку:

Цей код надає розв'язок до задачі, де задано натуральне число n та цілі числа a_1, \dots, a_n . Він перевіряє, чи послідовність впорядкована за зростанням або спаданням. Якщо послідовність є впорядкованою, то вона повертається без змін. Якщо послідовність не є впорядкованою, то вона модифікується, залишаючи лише підпослідовність до елемента зі зміною тенденції.

Блок-схема:



3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

Код:

```
#include <iostream>
#include <vector>
#include <algorithm>
```

```
int main() {
    // Введення кількості чисел
    std::cout << "Введіть кількість чисел: ";
    int n;
    std::cin >> n;

    // Введення послідовності чисел
    std::vector<int> numbers;
    std::cout << "Введіть послідовність чисел: ";
```

```

for (int i = 0; i < n; ++i) {
    int num;
    std::cin >> num;
    numbers.push_back(num);
}

// Перевірка чи послідовність вже впорядкована
bool isSortedAscending = std::is_sorted(numbers.begin(), numbers.end());
bool isSortedDescending = std::is_sorted(numbers.begin(), numbers.end(),
std::greater<int>());

if (isSortedAscending) {
    std::cout << "Послідовність впорядкована." << std::endl;
    return 0;
} else if (isSortedDescending) {
    std::cout << "Послідовність впорядкована." << std::endl;
    return 0;
}

// Вибір способу сортування
std::cout << "Оберіть спосіб сортування (1 - зростання, 2 - спадання): ";
int sortOption;
std::cin >> sortOption;

// Сортування у вибраному порядку
if (sortOption == 1) {
    std::sort(numbers.begin(), numbers.end());
} else if (sortOption == 2) {
    std::sort(numbers.begin(), numbers.end(), std::greater<int>());
} else {

```

```

        std::cout << "Неправильний вибір способу сортування." << std::endl;
        return 1;
    }

    if (numbers.size() >= 2) {
        int lastIdx = numbers.size() - 1;
        int secondLastIdx = lastIdx - 1;
        if (sortOption == 1) {
            std::swap(numbers[lastIdx], numbers[secondLastIdx]);
        } else {
            std::swap(numbers[secondLastIdx], numbers[lastIdx]);
        }
    }

    // Виведення відсортованої та зміненої послідовності
    std::cout << "Відсортована послідовність: ";
    for (const auto& num : numbers) {
        std::cout << num << " ";
    }

    return 0;
}

```

4. Опис інтерфейсу програми (керівництво користувача)

Основний інтерфейс програми має наступну структуру:

1. Користувач вводить натуральне число n - кількість елементів послідовності.
2. Користувач вводить цілі числа a_1, \dots, a_n , розділяючи їх пробілами.
3. Програма перевіряє, чи є послідовність вже впорядкованою за зростанням або за спаданням. Якщо так, виводить повідомлення "Послідовність залишається без змін".

4. Якщо послідовність не є впорядкованою, програма знаходить підпослідовність a_1, \dots, a_m , де $m < n$, таку, що або $a_1 \leq a_2 \leq \dots \leq a_m$ та $a_m > a_{m+1}$, або $a_1 \geq a_2 \geq \dots \geq a_m$ та $a_m < a_{m+1}$.

5. Програма виводить підпослідовність a_1, \dots, a_m .

5. Опис тестових прикладів

```
Введіть кількість чисел: 5
Введіть послідовність чисел: 3
7
98
234
13000
Послідовність впорядкована.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Введіть кількість чисел: 7
Введіть послідовність чисел: 9000
5034
3000
789
345
7
1
Послідовність впорядкована.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Введіть кількість чисел: 4
Введіть послідовність чисел: 34
1
8
3
Оберіть спосіб сортування (1 - зростання, 2 - спадання): 1
Відсортована послідовність: 1 3 34 8

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Введіть кількість чисел: 5
Введіть послідовність чисел: 56
7
34
1
89
Оберіть спосіб сортування (1 - зростання, 2 - спадання): 2
Відсортована послідовність: 89 56 34 1 7

...Program finished with exit code 0
Press ENTER to exit console.
```

6. Аналіз помилок (опис усунення зауважень)

1. Помилка: Умова перевірки порядку сортування в коді неправильно сформульована. Опис усунення: Перевірте чи послідовність впорядкована за зростанням за допомогою " $a[i] < a[i+1]$ " та за спаданням за допомогою " $a[i] > a[i+1]$ ". Якщо обидві перевірки повертають "false", то послідовність неупорядкована.
2. Помилка: Не визначена змінна "m" перед використанням у коді. Опис усунення: Спочатку визначте змінну "m" та ініціалізуйте її значенням, наприклад, 0.
3. Помилка: Умова перевірки напрямку послідовності в коді неправильно сформульована. Опис усунення: Перевірте чи послідовність має спадний напрямок за допомогою " $am > am+1$ " або зростаючий напрямок за допомогою " $am < am+1$ ".

7. Висновки

У цій лабораторній роботі було розроблено консольну програму на мові програмування C++, яка включає одне індивідуальне завдання.

Завдання вимагало введення даних від користувача, обробки цих даних та виведення результату на екран. В процесі розробки, було важливо враховувати потенційні помилки, такі як некоректне введення даних, невірна обробка винятків та несправжня обробка умов. Додатково, блок-схема могла б бути використана для кращого розуміння логіки завдання та виявлення можливих помилок.

У разі виявлення помилок, важливо було їх виправити та перевірити програму на різних вхідних даних для впевненості в її коректності.

У цілому, ця лабораторна робота надала можливість поглибити розуміння основ програмування на мові C++, вирішити завдання та навчитися виявляти та виправляти можливі помилки в програмі.

