

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

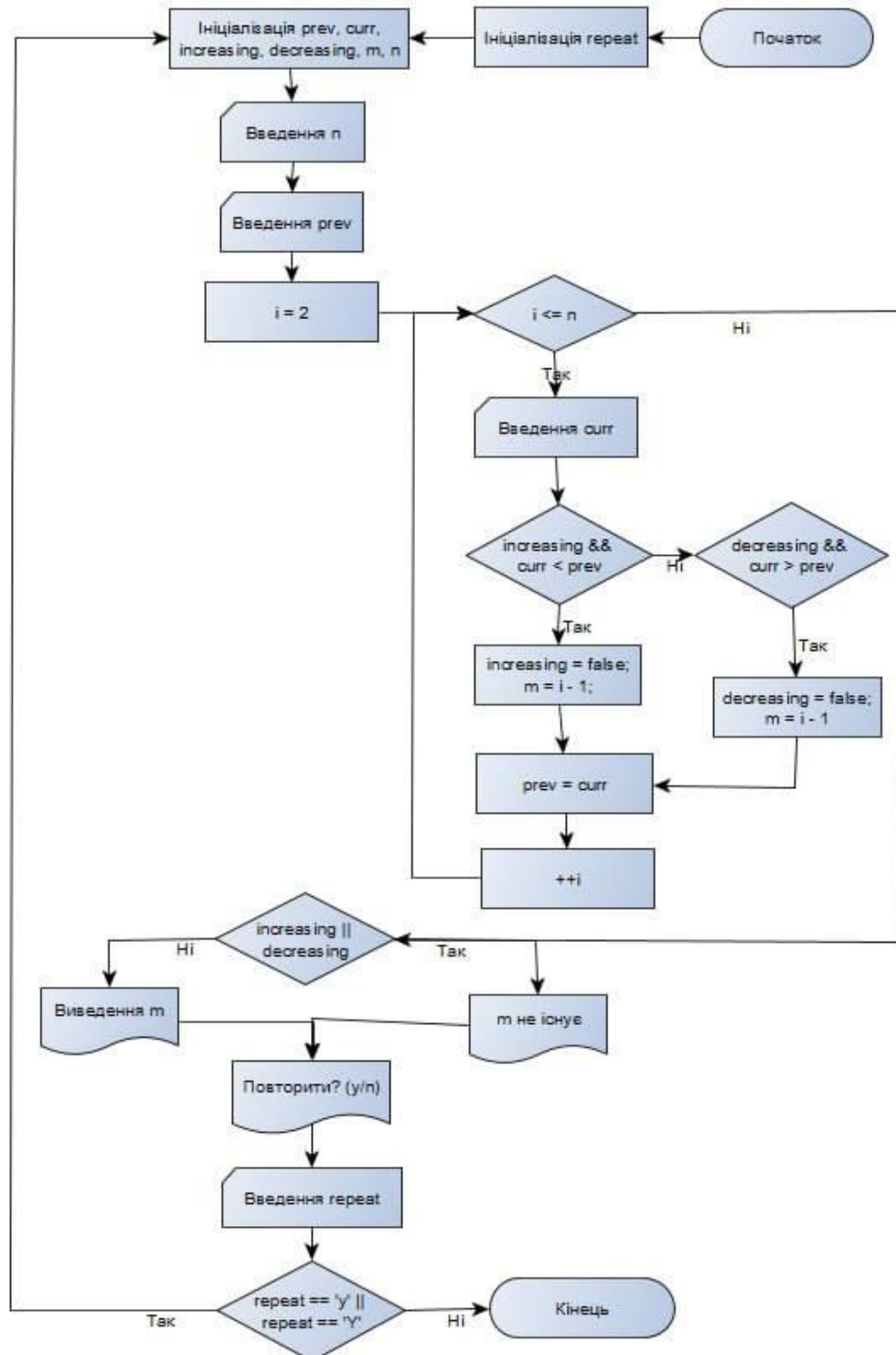
Опис розв'язку:

Цей фрагмент коду є розв'язком задачі, яка полягає у знаходженні значення змінної "m" у послідовності чисел, введених користувачем. Основна ідея полягає в тому, що ми перевіряємо, чи послідовність є лише зростаючою, лише спадаючою, або містить зміну напрямку. Основні кроки алгоритму:

1. Користувач вводить кількість членів послідовності (змінна "n").
2. Користувач вводить значення першого члена послідовності (змінна "prev").
3. Змінна "increasing" ініціалізується як true (припускаємо, що послідовність є зростаючою), і змінна "decreasing" ініціалізується як true (припускаємо, що послідовність є спадаючою).
4. Цикл від "i = 2" до "n" запитує у користувача значення поточного члена послідовності (змінна "curr").
5. Перевіряється, чи є зростанням та спаданням "curr" відносно попереднього елемента (змінна "prev"). Якщо перший нижче того, що попередній - змінні "increasing" та "decreasing" стають false, а змінна "m" оновлюється на значення "i - 1".
6. Поточний елемент стає попереднім елементом для наступної ітерації.
7. Після завершення циклу перевіряється, чи є послідовність лише зростаючою або лише спадаючою, і виводиться результат.

8. Користувачу пропонується повторити процес (за допомогою змінної "repeat").

Блок-схема:



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

Код:

```
#include <iostream>
```

```
int main() {
```

```
    char repeat;
```

```
    do {
```

```
        int n;
```

```
        std::cout << "Введіть кількість членів послідовності: ";
```

```
        std::cin >> n;
```

```
        int prev, curr;
```

```
        bool increasing = true, decreasing = true;
```

```
        int m = 1;
```

```
        std::cout << "Введіть значення для 1-го члена послідовності: ";
```

```
        std::cin >> prev;
```

```
        for (int i = 2; i <= n; ++i) {
```

```
            std::cout << "Введіть значення для " << i << "-го члена послідовності: ";
```

```
            std::cin >> curr;
```

```
            if (increasing && curr < prev) {
```

```
                increasing = false;
```

```
                m = i - 1; // Оновлюємо m при першому виявленні спадання
```

```
            }
```

```
            if (decreasing && curr > prev) {
```

```

        decreasing = false;
        m = i - 1; // Оновлюємо m при першому виявленні зростання
    }

    prev = curr;
}

if (increasing || decreasing) {
    std::cout << "m не існує" << std::endl;
}
else {
    std::cout << "Значення m: " << m << std::endl;
}

std::cout << "Бажаєте повторити? (y/n): ";
std::cin >> repeat;

} while (repeat == 'y' || repeat == 'Y');

return 0;
}

```

4. Опис інтерфейсу програми (керівництво користувача)

Цей код є прикладом простої програми на мові C++. Вона запитує користувача про кількість членів послідовності і вся послідовність введених чисел. Програма перевіряє, чи послідовність є лише зростаючою або лише спадаючою, і видає відповідне повідомлення. Якщо існує хоча б одна зміна напрямку (зростає, потім спадає), програма визначає значення *m* - номер попереднього члена послідовності, перед яким відбувся змiну напрямку.

Програма також дає можливість користувачеві повторити процес з іншою послідовністю.

5. Опис тестових прикладів

```
Введіть кількість членів послідовності: 5
Введіть значення для 1-го члена послідовності: 4
Введіть значення для 2-го члена послідовності: 78
Введіть значення для 3-го члена послідовності: 123
Введіть значення для 4-го члена послідовності: 11000
Введіть значення для 5-го члена послідовності: -567
Значення m: 4
Бажаєте повторити? (y/n): n

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

```
Введіть кількість членів послідовності: 3
Введіть значення для 1-го члена послідовності: 45
Введіть значення для 2-го члена послідовності: 7899
Введіть значення для 3-го члена послідовності: 1
Значення m: 2
Бажаєте повторити? (y/n): y
Введіть кількість членів послідовності: 4
Введіть значення для 1-го члена послідовності: 234
Введіть значення для 2-го члена послідовності: 456
Введіть значення для 3-го члена послідовності: 1
Введіть значення для 4-го члена послідовності: 45
Значення m: 2
Бажаєте повторити? (y/n): y
Введіть кількість членів послідовності: 2
Введіть значення для 1-го члена послідовності: 12
Введіть значення для 2-го члена послідовності: 56
m не існує
Бажаєте повторити? (y/n): n

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

```
Введіть кількість членів послідовності: 5
Введіть значення для 1-го члена послідовності: 3
Введіть значення для 2-го члена послідовності: 5
Введіть значення для 3-го члена послідовності: 7
Введіть значення для 4-го члена послідовності: 9
Введіть значення для 5-го члена послідовності: 11
m не існує
Бажаєте повторити? (y/n): n

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

```
Введіть кількість членів послідовності: 3
Введіть значення для 1-го члена послідовності: 3
Введіть значення для 2-го члена послідовності: 9
Введіть значення для 3-го члена послідовності: 2
Значення m: 2
Бажаєте повторити? (y/n): n

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

6. Аналіз помилок (опис усунення зауважень)

Опис усунення зауважень в програмі:

1. Додати пропущений заголовочний файл `<iostream>`, оскільки використовується об'єкт `std::cout` та `std::cin`.
2. Переконайтеся, що змінна `repeat` оголошена перед початком циклу `do-while`, оскільки вона використовується в умовному операторі циклу.
3. Переконайтеся, що змінні `prev` та `curr` оголошені перед циклом `for`, оскільки вони використовуються для зберігання попереднього та поточного значень членів послідовності.
4. Переконайтеся, що змінні `increasing` та `decreasing` оголошені перед циклом `for`, оскільки вони використовуються для перевірки чи послідовність є зростаючою або спадною.

5. Переконалися, що змінна `m` оголошена перед циклом `for`, оскільки вона використовується для зберігання індексу елемента, де відбувається зміна спадання або зростання.

6. Після введення кількості членів послідовності `n`, додати перевірку на значення. У випадку, якщо введена некоректна кількість, можна вивести повідомлення про помилку та вийти з програми.

7. Після введення значення для першого члена послідовності `prev`, додати перевірку на значення. У випадку, якщо введене некоректне значення, можна вивести повідомлення про помилку та повторити запит на введення.

8. Внутрішній цикл `for` можна замінити на `while`, оскільки потреба укласти попереднє і поточне значення в циклі необов'язкова. Можливо, це спростить програму та позбавить її від додаткових змінних.

9. Після циклу `do-while`, додати перевірку введеного значення для змінної `repeat`. У випадку, якщо введено некоректне значення, можна вивести повідомлення про помилку та завершити програму.

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.