**ЗВІТ**

**з дисципліни «Алгоритми та структури даних»**

**Лабораторне заняття № 1**

на тему: ««Однозв’язний список»

студента групи ПТ-23-1

Звонарьов Віктор

1. **Мета роботи**

Вивчення методів представлення зв’язаного списку в пам’яті ЕОМ; освоєння засобів мови програмування для реалізації операцій над однозв’язаними списками.

1. **Умова завдання**

**Завдання**

**Реалізувати структуру однозв’язаного списку. Для виконання операцій з елементами заданої структури створити функції, які реалізують:**

**- додавання елементу в початок списку;**

**- вставлення елементу після n-го елементу списку;**

**- пересування елемента на n позицій;**

**- видалення n-го елементу з списку;**

**- видалення кожного n-го елементу списку;**

**- впорядкувати елементи в списку за зростанням (спаданням);**

**- створення копії списку;**

**- очищення списку.**

**Створити два списки на основі розробленої структури однозв’язаного списку та створити функції, які реалізують:**

**- склеювання двох списків;**

**- створення списку, який містить спільні елементи двох списків.**

1. **Структура основних вхідних та вихідних даних**

У програмі використовується структура Node, яка містить наступні поля:

- data (int) — значення елемента списку.

- Node\* next — вказівник на наступний елемент списку.

У програмі використовується структура List, яка містить наступні поля:

- head (private Node\*) — вказівник на перший елемент списку.

Основні функції для роботи з елементами списку:

1. addToStart(int data) — додає новий елемент на початок списку.

Вхідні дані: число (значення нового елемента).

Вихідні дані: void.

2. deleteNth(int n) — видаляє елемент на позиції `n`.

Вхідні дані: номер елемента для видалення.

Вихідні дані: void або повідомлення про помилку, якщо елемент не знайдений.

3. moveBy(int n) — зміщує елементи списку на `n` позицій.

Вхідні дані: число `n` (кількість позицій для зміщення).

Вихідні дані: void.

4. sortAscending() і sortDescending() — сортує список за зростанням або спаданням.

Вхідні дані: немає.

Вихідні дані: void.

5. copy() — створює копію списку.

Вхідні дані: немає.

Вихідні дані: новий об'єкт типу `List` (копія списку).

6. concatenate(List \*other) — об'єднує поточний список з іншим списком.

Вхідні дані: вказівник на інший список.

Вихідні дані: void.

7. commonElements(List \*other) — знаходить спільні елементи між двома списками.

Вхідні дані: вказівник на інший список.

Вихідні дані: новий список, що містить спільні елементи.

8. clear() — очищає список.

Вхідні дані: немає.

Вихідні дані: void.

9. printList() — виводить список на екран.

Вхідні дані: вказівник на список.

Вихідні дані: void (друкує елементи списку).

10. main() — основний цикл, що забезпечує роботу інтерфейсу користувача з можливістю додавання, видалення та роботи з елементами списку.

**4. Текст програми (листинг)**

- addToStart(int data)

Функція додавання нового елемента на початок списку. Приймає значення нового елемента і додає його на початок списку.

- deleteNth(int n)

Функція для видалення елемента на позиції `n`. Якщо елемент знайдений, видаляється відповідний вузол. Якщо елемента на вказаній позиції не існує, виводиться повідомлення про помилку.

- moveBy(int n)

Функція зміщує елементи списку на `n` позицій. Якщо `n` більше довжини списку, нічого не відбувається.

- sortAscending() і sortDescending()

Функції для сортування списку за зростанням або спаданням.

Використовують метод бульбашкового сортування для впорядкування елементів.

- copy()

Функція створює копію поточного списку. Проходить по всіх елементах і додає їх у новий список.

- concatenate(List \*other)

Функція об'єднання двох списків. Приймає інший список і додає його елементи в кінець поточного списку.

- commonElements(List \*other)

Функція знаходить і повертає список спільних елементів між поточним списком та іншим.

-clear()

Функція очищає список, видаляючи всі його елементи.

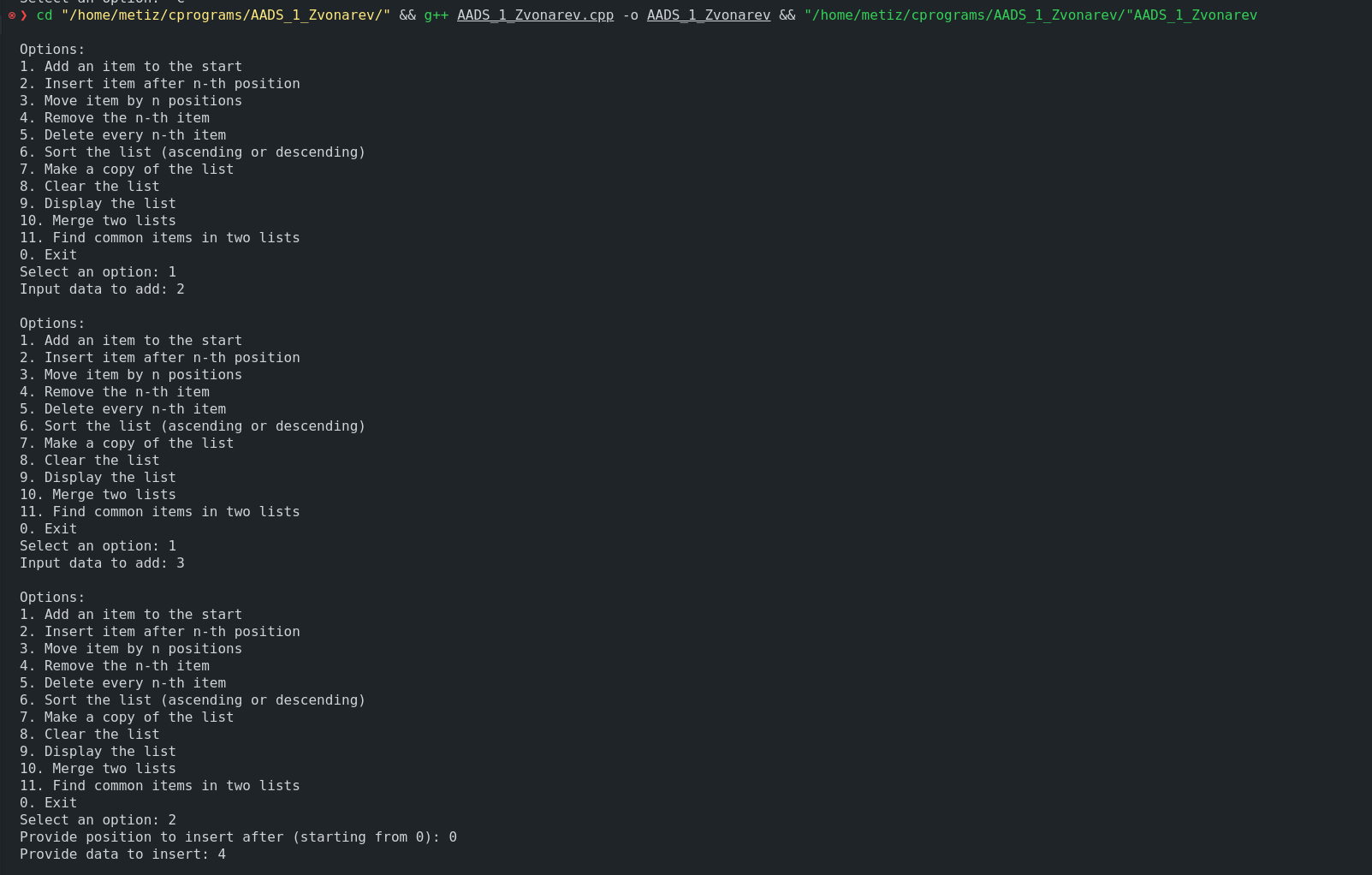
- printList()

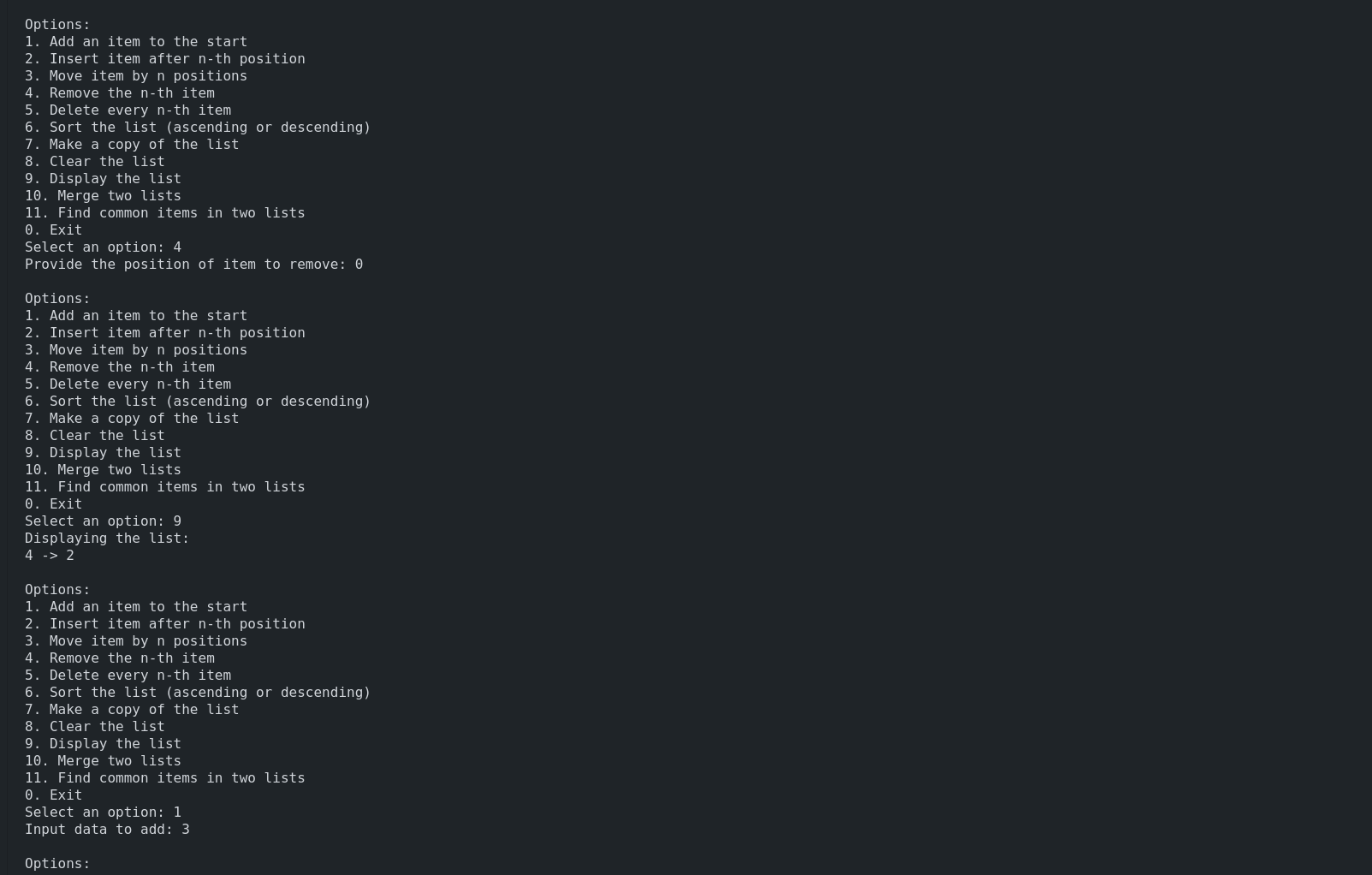
Функція для виведення всіх елементів списку на екран.

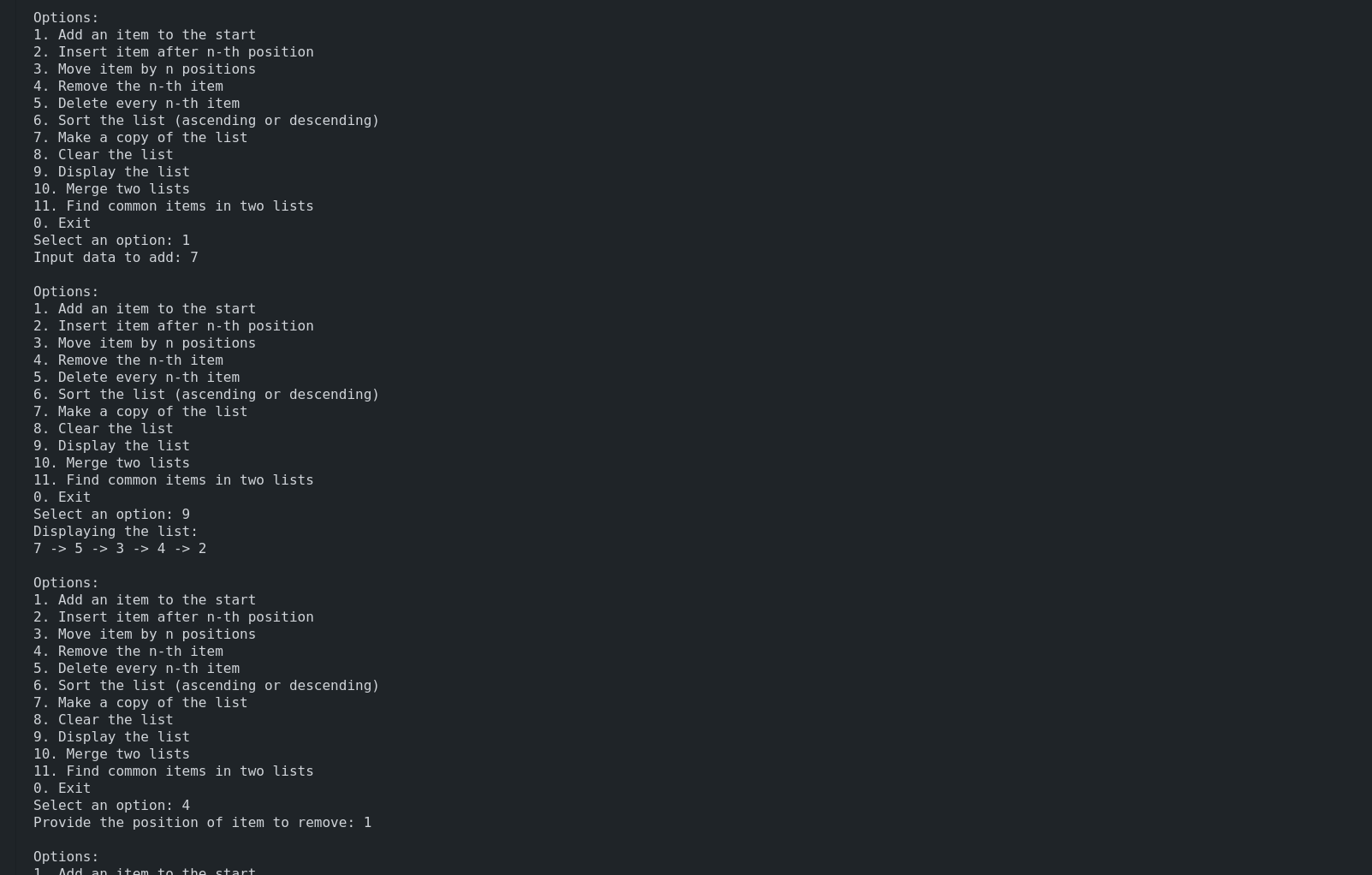
- main()

Основний модуль програми, який містить меню для взаємодії користувача з функціями додавання, видалення, сортування і виведення елементів списку. Програма завершує роботу при виборі користувачем опції "Exit".

1. **Набір тестів (Опис тестових прикладів)**







1. **Результати роботи програми та їх аналіз**

Програма успішно виконує основні операції з однозв’язним списком. Після введення даних, користувач може ефективно керувати списком, забезпечуючи простоту та зручність взаємодії через консольне меню.