**ЗВІТ**

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

**Лабораторне заняття № 4**

на тему: «Черга. Операції над чергою»

студента групи ПТ-23-1

Звонарьов Віктор

1. **Мета роботи**

Вивчення методів представлення черги в пам’яті ЕОМ; освоєння засобів мови програмування для реалізації операцій над чергою.

1. **Умова завдання**

**Завдання**

Реалізувати чергу. Для виконання операцій з елементами заданої структури створити функції, які додатково реалізують:

- основні операції (додавання та видалення елементу стеку);

- змінити місцями перший та останній елемент черги;

- розвернути чергу;

- операція «чи належить заданий елемент» черзі;

- видалити всі елементи черги.

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

У програмі використовується клас Queue, який містить такі поля:

- Node\* front — вказівник на початок черги.

- Node\* rear — вказівник на кінець черги.

Основні функції для роботи з елементами черги:

1. enqueue(int item) — додає новий елемент у стек. Якщо стек повний, автоматично збільшує його розмір удвічі.

- Вхідні дані: число (елемент для додавання).

- Вихідні дані: void.

2. dequeue() — видаляє перший доданий елемент з черги.

- Вхідні дані: немає.

- Вихідні дані: void.

3. swapFrontAndRear() — міняє місцями перший і останній елемент черги.

- Вхідні дані: немає.

- Вихідні дані: void.

4. reverse() — розвертає елементи у черзі, змінюючи порядок їх розташування на зворотний.

- Вхідні дані: немає.

- Вихідні дані: void.

5. contains(int item) — перевіряє, чи містить черга заданий елемент.

- Вхідні дані: число (елемент для пошуку).

- Вихідні дані: void (виводить повідомлення про наявність елемента).

6. clear() — очищає чергу, видаляючи всі елементи.

- Вхідні дані: немає.

- Вихідні дані: void.

7. print() — виводить всі елементи черги у поточному порядку.

- Вхідні дані: немає.

- Вихідні дані: void.

**4. Текст програми (листинг)**

1. enqueue(int item): додає новий елемент у чергу.

2. dequeue(): видаляє елемент з початку черги.

3. swapFrontAndRear(): міняє місцями перший і останній елементи черги.

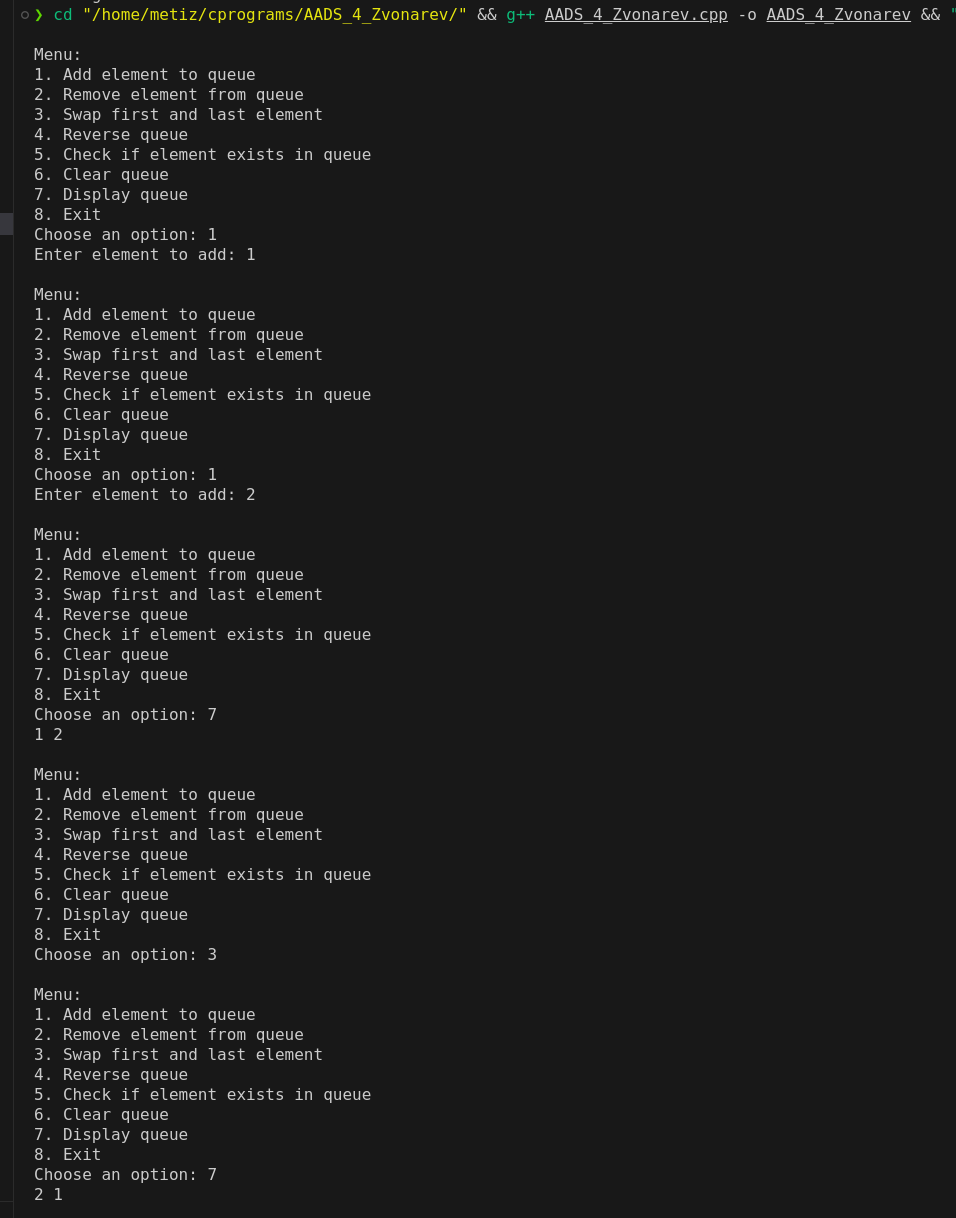
4. reverse(): реверсує порядок елементів у черзі.

5. contains(int item): перевіряє, чи містить черга заданий елемент.

6. clear(): очищає чергу, видаляючи всі елементи.

7. print(): виводить всі елементи черги у порядку їх додавання.

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







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

Під час виконання програми користувач взаємодіє з меню, яке дозволяє виконувати різноманітні операції над чергою, такі як додавання, видалення елементів, реверсування, пошук елементів тощо.