1. **Tạo Node từ giá trị nguyên (createNode)**
   - **Cách làm:**
     - **Cấp phát bộ nhớ cho Node mới**
     - **Gán giá trị data vào trường key**
     - **Đặt con trỏ p_next trỏ tới nullptr**
2. **Tạo List từ Node (createList)**
   - **Cách làm:**
     - **Tạo List mới**
     - **Nếu Node đầu vào khác nullptr, gán cả head và tail trỏ vào Node đó**
     - **Ngược lại, khởi tạo List rỗng (head = tail = nullptr)**
3. **Thêm vào đầu (addHead)**
   - **Cách làm:**
     - **Tạo Node mới từ giá trị data**
     - **Nếu List rỗng: head và tail cùng trỏ vào Node mới**
     - **Ngược lại: Node mới trỏ vào head hiện tại, sau đó cập nhật head**
4. **Thêm vào cuối (addTail)**
   - **Cách làm:**
     - **Tạo Node mới**
     - **Nếu List rỗng: head và tail cùng trỏ vào Node mới**
     - **Ngược lại: tail hiện tại trỏ vào Node mới, cập nhật tail mới**
5. **Xóa đầu (removeHead)**
   - **Cách làm:**

- Nếu List rỗng → trả về false
- Nếu chỉ có 1 Node: xóa Node và trả tail về nullptr
- Ngược lại: lưu head hiện tại, di chuyển head tới Node kế tiếp, xóa Node cũ

6. Xóa cuối (removeTail)
   - Cách làm:
     - Nếu List rỗng → không làm gì
     - Nếu có 1 Node: xóa Node, set head và tail về nullptr
     - Ngược lại: duyệt từ đầu tìm Node trước tail, xóa tail, cập nhật tail mới
7. Xóa toàn bộ (removeAll)
   - Cách làm:
     - Duyệt từ đầu List
     - Lần lượt xóa từng Node cho đến khi List rỗng
     - Cuối cùng set tail về nullptr
8. Xóa Node trước giá trị (removeBefore)
   - Cách làm:
     - Tìm Node có giá trị val
     - Xóa Node ngay trước nó
     - Cần lưu Node trước đó (beforePrev) và Node cần xóa (prev)
9. Xóa Node sau giá trị (removeAfter)
   - Cách làm:
     - Tìm Node có giá trị val
     - Nếu Node này có Node sau → xóa Node sau
     - Cập nhật con trỏ p_next và tail nếu cần

10. Thêm vào vị trí (addPos)

- **Cách làm:**
  - **Nếu pos = 0 → dùng addHead**
  - **Duyệt tới vị trí pos-1**
  - **Chèn Node mới vào sau vị trí này**
  - **Cập nhật tail nếu chèn vào cuối**

11. Xóa tại vị trí (removePos)

- **Cách làm:**
  - **Nếu pos = 0 → dùng removeHead**
  - **Duyệt tới vị trí pos-1**
  - **Xóa Node ở vị trí pos**
  - **Cập nhật con trỏ và tail nếu cần**

12. Thêm trước giá trị (addBefore)

- **Cách làm:**
  - **Tìm Node có giá trị val**
  - **Nếu là head → dùng addHead**
  - **Ngược lại chèn Node mới vào trước Node tìm được**

13. Thêm sau giá trị (addAfter)

- **Cách làm:**
  - **Tìm Node có giá trị val**
  - **Chèn Node mới vào sau Node tìm được**
  - **Cập nhật tail nếu chèn vào cuối**

14. In List (printList)

- **Cách làm:**
  - **Duyệt từ đầu đến cuối List**
  - **In giá trị từng Node, ngăn cách bởi "->"**
  - **Kết thúc bằng "nullptr"**

15. Đếm phần tử (countElements)

- **Cách làm:**
  - **Khởi tạo biến đếm = 0**
  - **Duyệt List, tăng biến đếm với mỗi Node**
  - **Trả về giá trị đếm**

16. Đảo ngược List (reverseList)

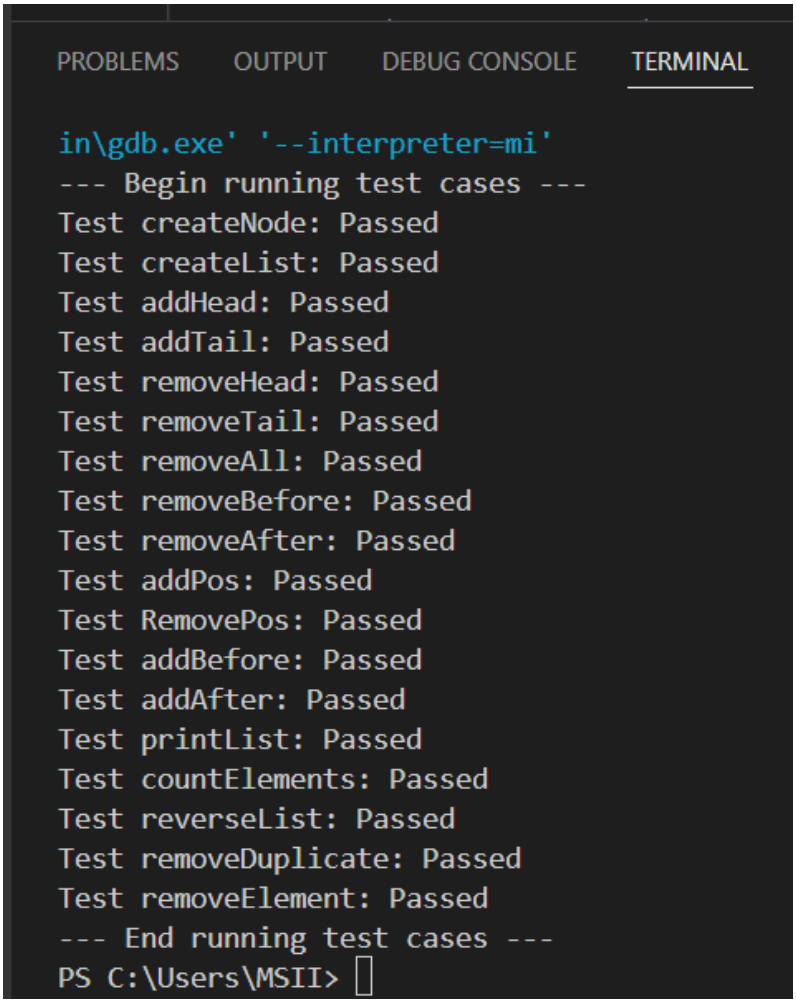
- **Cách làm:**
  - **Tạo List mới**
  - **Duyệt List cũ từ đầu đến cuối**
  - **Lần lượt thêm các Node vào đầu List mới**

17. Xóa trùng lặp (removeDuplicate)

- **Cách làm:**
  - **Duyệt List**
  - **Với mỗi Node, kiểm tra và xóa các Node sau nó có cùng giá trị**
  - **Chú ý cập nhật tail khi xóa Node cuối**

18. Xóa phần tử (removeElement)

- **Cách làm:**
  - **Tìm Node có giá trị key**
  - **Nếu là head → removeHead**
  - **Nếu là tail → removeTail**
  - **Ngược lại xóa Node và nối các Node trước/sau với nhau**



1. **Tạo Node từ giá trị nguyên (createNode)**
   - **Cách làm:**
     - **Cấp phát bộ nhớ cho Node mới**
     - **Gán giá trị data vào trường key**
     - **Đặt con trỏ p_next trỏ tới nullptr**
   - **Lưu ý: Nhớ kiểm tra cấp phát bộ nhớ thành công**
2. **Tạo List từ Node (createList)**
   - **Cách làm:**

- Tạo List mới
- Nếu Node đầu vào khác nullptr, gán cả head và tail trỏ vào Node đó
- Ngược lại, khởi tạo List rỗng (head = tail = nullptr)

3. Thêm vào đầu (addHead)
   - Cách làm:
     - Tạo Node mới từ giá trị data
     - Nếu List rỗng: head và tail cùng trỏ vào Node mới
     - Ngược lại: Node mới trỏ vào head hiện tại, sau đó cập nhật head
4. Thêm vào cuối (addTail)
   - Cách làm:
     - Tạo Node mới
     - Nếu List rỗng: head và tail cùng trỏ vào Node mới
     - Ngược lại: tail hiện tại trỏ vào Node mới, cập nhật tail mới
5. Xóa đầu (removeHead)
   - Cách làm:
     - Nếu List rỗng → trả về false
     - Nếu chỉ có 1 Node: xóa Node, set head và tail về nullptr
     - Ngược lại: lưu head hiện tại, di chuyển head tới Node kế tiếp, xóa Node cũ
6. Xóa cuối (removeTail)
   - Cách làm:

- Nếu List rỗng → không làm gì
- Nếu có 1 Node: xóa Node, set head và tail về nullptr
- Ngược lại: duyệt từ đầu tìm Node trước tail, xóa tail, cập nhật tail mới

7. Xóa toàn bộ (removeAll)

- Cách làm:
  - Duyệt từ đầu List
  - Lần lượt xóa từng Node cho đến khi List rỗng
  - Cuối cùng set tail về nullptr

8. Xóa Node trước giá trị (removeBefore)

- Cách làm:
  - Tìm Node có giá trị val
  - Xóa Node ngay trước nó
  - Cần lưu Node trước đó (beforePrev) và Node cần xóa (prev)

9. Xóa Node sau giá trị (removeAfter)

- Cách làm:
  - Tìm Node có giá trị val
  - Nếu Node này có Node sau → xóa Node sau
  - Cập nhật con trỏ p_next và tail nếu cần

10. Thêm vào vị trí (addPos)

- Cách làm:
  - Nếu pos = 0 → dùng addHead
  - Duyệt tới vị trí pos-1
  - Chèn Node mới vào sau vị trí này
  - Cập nhật tail nếu chèn vào cuối

11. Xóa tại vị trí (removePos)

- **Cách làm:**
  - **Nếu pos = 0 → dùng removeHead**
  - **Duyệt tới vị trí pos-1**
  - **Xóa Node ở vị trí pos**
  - **Cập nhật con trỏ và tail nếu cần**

12. Thêm trước giá trị (addBefore)

- **Cách làm:**
  - **Tìm Node có giá trị val**
  - **Nếu là head → dùng addHead**
  - **Ngược lại chèn Node mới vào trước Node tìm được**

13. Thêm sau giá trị (addAfter)

- **Cách làm:**
  - **Tìm Node có giá trị val**
  - **Chèn Node mới vào sau Node tìm được**
  - **Cập nhật tail nếu chèn vào cuối**

14. In List (printList)

- **Cách làm:**
  - **Duyệt từ đầu đến cuối List**
  - **In giá trị từng Node, ngăn cách bởi "->"**
  - **Kết thúc bằng "nullptr"**

15. Đếm phần tử (countElements)

- **Cách làm:**
  - **Khởi tạo biến đếm = 0**
  - **Duyệt List, tăng biến đếm với mỗi Node**
  - **Trả về giá trị đếm**

16. Đảo ngược List (reverseList)

- **Cách làm:**

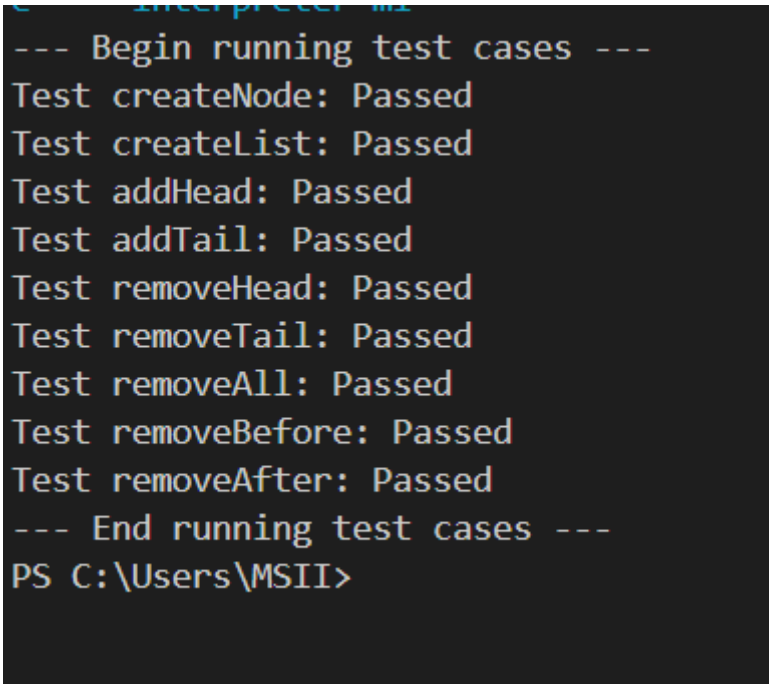
- Tạo List mới
- Duyệt List cũ từ đầu đến cuối
- Lần lượt thêm các Node vào đầu List mới

17. Xóa trùng lặp (removeDuplicate)

- Cách làm:
  - Duyệt List
  - Với mỗi Node, kiểm tra và xóa các Node sau nó có cùng giá trị
  - Chú ý cập nhật tail khi xóa Node cuối

18. Xóa phần tử (removeElement)

- Cách làm:
  - Tìm Node có giá trị key
  - Nếu là head → removeHead
  - Nếu là tail → removeTail
  - Ngược lại xóa Node và nối các Node trước/sau với nhau


--- Begin running test cases ---
Test createNode: Passed
Test createList: Passed
Test addHead: Passed
Test addTail: Passed
Test removeHead: Passed
Test removeTail: Passed
Test removeAll: Passed
Test removeBefore: Passed
Test removeAfter: Passed
--- End running test cases ---
PS C:\Users\MSII>