Lab 2: Singly-linked List

You may work on this exercise with another student. If you do, make sure you write **both names in the header**.

Complete the given project by implementing the following functions in the class AnyList:

- **Overloaded insertion operator**
  - Prints the list in the following format, starting from the element at the front of the list:
    elem1 elem2 elem3…
- **Constructor**
- **Copy constructor**
- Function **insert**
  - **Parameter:** An int storing the element to insert
  - After inserting the first element in an empty list, inserts the second element at the end of the list, the third element to the front of the list, the fourth element at the end of the list, and so on.
    Example:
    Input numbers (given in this order): 1 2 3 4 5 6 7
    Resulting list: (front) 7 5 3 1 2 4 6 (back)
- **Overloaded assignment operator**
- **Function clear**
  - Re-sets the list to an empty list.
- **Destructor**

In the **Main.cpp** file, create testing cases for the following lists:

- (empty list)
- List: 33
- List: 5 3
- List: 9 6 8
- List: 4 9 1
- List: 7 5 9 3 2 1 4 6 8
- List: 4 5 3 6 2 7 1 8