Create a class called LinkedList using the specifications as discussed in class including the .h and .cpp files.

Use the following struct:

```cpp
struct Node
{
    int val;
    Node * next;
};
```

Use the following structure for your class:

```cpp
class LinkedList
{
    Node * head;
    Public: \ \ methods
        LinkedList \ \ \ \ \ \ constructor
        PrintList \ \ \ \ \ \ prints elements of the list one per line
        \ \ \ if list is empty, print function should print “Printing empty list”
        Insert_Front \ \ \ \ \ \ inserts element to front of list
        Insert_Rear \ \ \ \ \ \ inserts element to end of list
        Insert_Ordered \ \ \ \ \ \ inserts element in numerical order – call Search
        Search \ \ \ \ \ \ searches for an element in the list;
        \ \ \ \ \ \ \ \ \ \ \ input parameter is value you are searching for
        \ \ \ \ \ \ \ \ \ \ \ Returns True or False if found or not through return statement
        \ \ \ \ \ \ \ \ \ \ \ Returns through parameter the pointer to the item if found or
        \ \ \ \ \ \ \ \ \ \ \ the previous item if not found (Null if empty or at front)
        Delete_Node \ \ \ \ \ \ removes an element from the list – call Search
        Clear_list \ \ \ \ \ \ resets the LinkedList to empty
};
```

Write a main function to test your class. A file will be provided to use for turning in. It will be of the following form – single character designating the operation, integer indicating data if necessary, 0 will be a dummy argument for consistency. Use the data stream below for turning in.

In `main `, print a comment for each function call. For example,

**Printing list**

Inserting 50 to ordered list
Inserting 79 to front of list … etc.

```cpp
P 0 // Print the (empty) list
O 50 // Insert 50 to an ordered list
R 79 // Insert 79 to the front of the list
F 34 // Insert 34 to the rear of the list
O 55 // Insert 55 to the list in numerical order
O 90 // Insert 90 to ordered list
S 79 // Search the list for 79 – print comment if found or not
D 50 // Delete node containing 50 from the list
D 62 // Delete 62
P 0 // Print list
C 0 // Clear the list
P 0 // Print cleared final list
```

You will be given an application to use the class for your next project.