Class name:
We are implementing two classes called ItemType and UnsortedType.

ItemType Class Functions:
The class ItemType will have the following functions:

1. RelationType ComparedTo(ItemType) const;
2. void Print(std::ostream&) const;
3. void Initialize(int number);

UnsortedType Class Functions:
The class UnsortedType will have the following functions:

1. void MakeEmpty();
2. bool IsFull() const;
3. int GetLength() const;
4. ItemType GetItem(ItemType& item, bool& found);
5. void PutItem(ItemType item);
6. void DeleteItem(ItemType item);
7. void ResetList();
8. ItemType GetNextItem();

Description of the ItemType functions:
1. ComparedTo(ItemType) const: This function compares the value with another item
2. Print (std :: ostream&) const: This function prints out the value of the ItemType
3. Initialize (int number): This function initializes the value to the number that is inputted.

Description of the UnsortedType functions:
1. MakeEmpty(): This function returns the list to the empty state.
2. IsFull() const: This function determines whether the list is full or not.
3. GetLength() const: This function determines the number of elements in the list.
4. GetItem(ItemType& item, bool& found): Retrieves list element whose key matches the item's key (if present)
5. PutItem(ItemType item): This function adds items to the list.
6. DeleteItem(ItemType item): This function deletes items from the list.
7. ResetList(): This function initializes the current position for an iteration through the list.
8. GetNextItem(): This function gets the next element in the list.

Test cases for ItemType functions:
1. ComparedTo(ItemType) const has **two** possible test case(s):
    a. The value returned will be LESS

   b. The value returned will be GREATER
   c. The value returned will be EQUAL
2. Print (std :: ostream&) const has **two** possible test case(s):
   a. The value of the ItemType will be printed to the stream out.
   b. The value of the ItemType will not be printed to the stream out.
3. Initialize (int number) has **two** possible test case(s):
   a. The value will be initialized to the int number.
   b. The value will not be initialized to the int number.

Test cases for UnsortedType functions:

1. void MakeEmpty(): has **two** possible test case(s):
   a. The list will be made empty.
   b. The list will not be made empty.
2. bool IsFull() const: has **two** possible test case(s):
   a. The list will be full.
   b. The list will not be full.
3. int GetLength() const has **two** possible test case(s):
   a. The length of the list will be returned.
   b. The length of the list will not be returned.
4. ItemType GetItem(ItemType& item, bool& found) has **two** possible test case(s):
   a. The item is found.
   b. The item is not found.
5. void PutItem(ItemType item) has **two** possible test case(s):
   a. The item will be added to the list.
   b. The item will not be added to the list.
6. void DeleteItem(ItemType item) has **one** possible test case(s):
   a. The item will be deleted.
7. void ResetList() has **two** possible test case(s):
   a. The list will be reset.
   b. The list will not be reset.
8. ItemType GetNextItem() has **two** possible test case(s):
   a. The next item in the list will be returned.
   b. The next item in the list will not be returned.

Program Test Cases:

Test case 1: Have all the different commands in an input file to see if they all work.
Test case 2: Adding numbers to a list and emptying the list multiple times before adding a final number and outputting it.
Test case 3: Adding numbers till the list is full then trying to fill the list after it's full.
Test case 4: Adding numbers then deleting them all.

Test case 5: Performing all the commands in different order multiple times in one program run then quitting.