CS3323
Project 1

In this project, you will implement an unsorted list based on **linked structure**. The program consists of two classes, **UnsortedType** and **ItemType**. (if you feel you need more classes, you create your own class to implement the project.) The logical description of each class is as follows:

1. **UnsortedType**
   a. Contents (data) of the list: The list elements are of ItemType.
   b. Operations (Some of them are exactly the same the ones in the book. You can copy them directly from the book if you decided to do so.)
      i. void PutItem (ItemType item)
         - This function puts new data (item) into the list.
         - The new item should be inserted at the end of the list.
         - If the requested data is already in the list, your program should report an error message. This doesn't mean that you program halts; rather it should continue until all inputted operations are executed.
      ii. boolean CheckItem (ItemType item)
         - This function checks if the requested data (item) is in the list.
         - If it doesn't exist in the list, your program should report an error message. This doesn't mean that you program halts; rather it should continue until all inputted operations are executed.
      iii. void DeleteItem (ItemType item)
         - This function deletes the requested data (item) from the list.
         - If it doesn't exist in the list, your program should report an error message. This doesn't mean that you program halts; rather it should continue until all inputted operations are executed.
      iv. void SplitList (UnsortedType& list, ItemType item, UnsortedType& list1, UnsortedType& list2)
         - Divides the list into two lists according to the key of the item.
         - list1 contains all the items of list whose keys are less than or equal to item's key; list2 contains all the items of list whose keys are greater than items key.
         - After splitting the list, both lists have to be displayed.
      v. void MakeEmpty( )
         - Make the entire list empty. (all items have to be deallocated.)
      vi. void PrintList( ) const
         - This function is used print all of the contents of the list.
      vii. void ResetList( )
         - Initialize current position of an iteration through the list.
      viii. ItemType GetNextItem( )
         - Gets the next element in the list. (Remember that current position should be updated before the function finishes execution.)
2. **ItemType**
   a. This class contains the following components:
      i. enum RelationType {LESS, GREATER, EQUAL};
      ii. member variable with "private" access modifier.
      iii. All operations described in section b below.
   b. Operations
      i. void ItemType()
         - Constructor of the ItemType class
      ii. RelationType CompareTo (ItemType otherItem) const

- This compares key data of the ItemType objects return one of the enum types defined earlier.

iii. void Initialize (int number)

- This function initializes key data of the ItemType objects.

To run your program, there will be an input file that contains the sequence of operations. When you program gets started, it should ask its user the input file name so that your program reads the contents of that input file. The following is input-output examples.

Input File

```
PutItem 24
PutItem 31
PutItem 1
PutItem 3
PutItem 5
PutItem 13
PutItem 18
CheckItem 3
DeleteItem 17
DeleteItem 31
PutItem 24
CheckItem 5
PrintList
ResetList
GetNextItem
GetNextItem
CheckItem 33
GetNextItem
SplitList 5
MakeEmpty
PrintList
```

```
PutItem 24
24
PutItem 31
24 31
PutItem 1
24 31 1
PutItem 3
24 31 1 3
PutItem 5
24 31 1 3 5
PutItem 13
24 31 1 3 5 13
PutItem 18
24 31 1 3 5 13 18
CheckItem 3
Requested data "3" is in the list.
24 31 1 3 5 13 18
DeleteItem 17
Requested data "17" is not in the list.
24 31 1 3 5 13 18
DeleteItem 31
24 1 3 5 13 18
PutItem 24
Requested data "24" is already in the list.
24 1 3 5 13 18
CheckItem 5
Requested data "5" is in the list.
24 1 3 5 13 18
```

Output File

```
PrintList
24 1 3 5 13 18
ResetList
Current positions has been reset.
24 1 3 5 13 18
GetNextItem
24
24 1 3 5 13 18
GetNextItem
1
24 1 3 5 13 18
CheckItem 33
Requested data "33" is not in the list
24 1 3 5 13 18
GetNextItem
3
24 1 3 5 13 18
SplitList 5
List 1: 1 3 5, List 2: 24 13 18
24 1 3 5 13 18
MakeEmpty
List became empty.
PrintList
List is empty.
```

Submission Requirements

- You must submit the program that compiles and runs without any error to receive full points.
- Along with the source code, please submit "ReadMe" file in text format, describing about your program; the IDE used, hardware/software environment (e.g. MacBook with i5, Windows OS, etc…), special instruction to run the code, etc.
- Don't forget to put comments in your source code so that the reader can understand your program easily.
- On the top of each source code file, please don't forget to put your Name.