Lab 13

In this lab, we will practice coding unsorted linked list. This program reads in a number of values from the data file (grocery.dat) and builds an unsorted list using these values. The format of the data file is as the following:

apple
orange
pineapple
milk
...

Write the program with the components as shown in the program structure below:

```
#include <fstream>
#include <iostream>
#include <string>
#include <cassert>
using namespace std;
typedef string ItemType;
struct NodeType;
typedef NodeType* NodePtr;
struct NodeType {
  ItemType data;
  NodePtr next;
};
void BuildList(ifstream & myIn, NodePtr & head);
void DislayList(NodePtr head);
bool IsPresent(NodePtr head, string item);
int main()
{
   ifstream myIn;
   NodePtr  head;  // the pointer points to the beginning of the list

   myIn.open("grocery.dat");
   assert(myIn);

   // call the function "BuildList" to read data from the data file and construct the list

   // call the function "DisplayList" to display the list

   //cout << "Enter an item to search for:"
   // call the function "IsPresent" to determine if the item is in the list
   // If the item is in the list,
   //     display a message "Item is found in the list"
   // else
   //     display a message "The item is not in the list"
   myIn.close();
```

```
    return 0;
}

// Define the function "BuildList" here.
// Write a "while" loop to read items from the data file, one item at a time.
// For each item read, add it at the end of the list.
// The loop terminates when the end of the data file is reached.
void BuildList(ifstream & myIn, NodePtr & head)
{
}

// Define the function "DisplayList" here.
void DislayList(NodePtr head)
{
}

// Define the function "IsPresent" here. This function returns true if the item to search for
is
// in the list, otherwise it returns false
bool IsPresent(NodePtr head, string item)
{
}
```

Test the program

Run the program 3 times, search for an item in the middle of the list. The second time, search for an item at the end of the list. The third time, search for an item that is not in the list.