Lab 05 - Linked Lists

Instructions:

- **The lab requires completing a few tasks.**
- **Your submissions must be submitted to the Lab05 directory of your GitHub repository or uploaded to the Lab05 assignment on Google classroom.**
- **Accompanying these instructions are a few header files that must be included in the appropriate programs you have to write.**
- **Besides the header files provided, your programs can only include the libraries *iostream*, *string*, *fstream*, *sstream*, and *cctype*.**
- **Cheating of any kind is prohibited and will not be tolerated.**
- **Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the lab.**

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, AT THE BEGINNING OF YOUR SUBMISSION(S), ADD A COMMENT THAT CONSISTS OF YOUR NAME AND THE DATE

Grading:

Task	Maximum Points	Points Earned
1	10	
Total	10	

Note: solutions will be provided for tasks colored blue only.

Task 1

In the file `"main.cpp"`, define

a. **an int function named `AboveValues()` whose header is**

```
int AboveValues(Node<int>* root,int target)
```

Given that *root* points to the head of the linked list, if the linked list is not empty, the function returns the number of values in the linked list that are more than or equal to *target*; otherwise, it returns 0.

b. **a bool *Node* pointer function named `FillList()` whose header is**

```
Node<bool>* FillList(int n)
```

If *n* is greater than or equal to 1, the function creates a linked list of length *n*, assigns false to each node, and then, returns the linked list; otherwise, it returns null.

c. **a void function named `MakeCircular()` whose header is**

```
template <typename T>
void MakeCircular(Node<T>*& root)
```

Given that *root* points to the head of the linked list, if the linked list is not empty, the function turns the linked list into a circular linked list; otherwise, it does nothing.

d. **a bool function named `IsNotMember()` whose header is**

```
template <typename T>
bool IsNotMember(Node<T>* root, Node<T>* node)
```

Given that *root* points to the head of the linked list, the function returns true if the linked list is not empty and *node* does not point to a node of the linked list; otherwise, it returns false. Furthermore, if *node* is equal to null, it returns true.

e. **a bool function named `NotContained()` whose header is**

```
bool NotContained(Array<int>& data,Node<T>* root)
```

Given that *root* point to the head of the linked list, the function returns true if the linked list pointed to by *root* contains no values in *data*; otherwise, it returns false. For instance, the callers `NotContained([1,2,4,4],[3,5,6])` and `NotContained([2,4],[1,2,3,5])` will return `true` and `false` respectively.