Question: Write a program that have a linked list that can convert to circular linked list and get back to singly when necessary methods called. Each node stores another linked list.

You will create a class called **ConvertableLinkedList**.
It has 2 **private** data members.

- **list** is a **singly linkedlist** that **each node stores an another linkedlist** which **each of the nodes** are **generic types**.
- **current** is a **node type** data member that points (or stores) the current node,

Also it has 2 **private** member functions called **convertToCircular()** and **convertToSingly()**, and 4 **public** member functions called **insertToLast(), insertAtIndex(), findNext()** and **display()**.

void convertToCircular(): It's a void function. Takes no parameters. Converts list to a circular linkedlist

void convertToSingly(): It's a void function. Takes no parameters. Converts back list from circular to singly

void insertToLast(const T&): It's a void function. Takes a **generic type value** that **same type with the linkedlist's nodes** as parameter, and insert it to the end of the list with a new linkedlist (remember that each node stores another linked list, so this parameter should be inserted to new linkedlist and you are going to insert that linkedlist to your list data member)

void insertAtIndex(const T&, int): It's a void function. Takes a **generic type value** that **same type with the linkedlist's nodes** as parameter and an **index number**. It inserts the parameter data to linkedlist that found at parameter index on the list data member. If the index number parameter is greater than the length of the linkedlist, it inserts the parameter value to the end of the linkedlist at the last node.

int findNext(const T&): Returns an integer. It's a **convert necessary function**. It takes a **generic type value** that **same type with the linkedlist's nodes** as parameter. It will search the parameter value in the linkedlists that found on the nodes of the list, **starting from the current node**. If it found **returns the index number** and end the search, else it returns **-1**.

void display(): It's a void function. Takes no parameters. Print list to the console as in sample run.

In main function, create a **ConvertableLinkedList** object and show the usage of each public member functions.

* **convert necessary function:** A function which needs circular linkedlist. Inside this function's body you need to convert the list to a circular linked list with using **convertToCircular()** function at the beginning. At the end of the function you need to convert it back to the singly linkedlist with using **convertToSingly()** function.

! Use the **linkedlist.h** header file that we provide it to you. **Do not make changes** in the header file.

<u>Sample Run:</u>

```
Case 1-Insert value to last
Case 2-Insert value to the Index
Case 3-Fınd the value
Case 4-Display
Case 5-Quit

Select a case: 1
Enter the value: 5
Enter your selection: 4
List printed:
5

Select a case: 1
Enter the value: 6
Enter your selection: 4
List printed:
5
6

Select a case: 2
Enter the value: 4
Enter the index: 0
Enter your selection: 4
List printed:
5 4
6

Select a case: 3
Enter the value you want to search: 4
Your value found in index 0

Select a case: 5
Goodbye
```