

Objectives:

After performing this lab, students shall be able to revise:

- Doubly Link List
- Circular link list

Question 1:

Implement a class 'Node' that contains two data members: A template variable 'data', Node pointer 'next'. Now implement a double link list linked list class having two private data members Node pointer 'tail' and 'head' Create basic functions like constructors, insert, delete and destructor according to your need. Also implement the following functions:

1. ReverseLinkedList

Given the head of a double link list you are required to reverse that link list and then return the reversed link list. You are not allowed to create another link list or simply exchange the data between nodes.

Example:

Input: 1, 2, 3, 4, 5

Output: 5, 4, 3, 2, 1

2. IsPalindrome

Given the head of a double link list you are required to check if it is a palindrome or not assuming that the link list is even.

Example 1:

Input: 3, 5, 5, 3

Output: Palindrome

Example 2:

Input: 1, 2, 3, 1

Output: Not a Palindrome

3. ReorderLinkedList

Given the head of a double link list of the order $L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$. Reorder the link list in the format $L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$

The nodes are required to be reordered not the node data.

Example 1:

Input: 1, 2, 3, 4, 5

Output: 1, 5, 2, 4, 3

Example 2:

Input: 6, 7, 8, 9

Output: 6, 9, 7, 8

4. SwapNodes

Given the head of a double link list and an integer 'n'. Swap the node at n^{th} position from the start of the link list with the n^{th} node from the end of the link list. Do not swap the data of the nodes. Do not use tail.

Example 1:

Input: 1, 2, 3, 4, 5, 6, 7, 8, 9 and $n=3$

Output: 1, 2, 7, 4, 5, 6, 3, 8, 9

Explanation: the 3rd element from the start is 3 and the 3rd element from the end is 7 so 3 is swapped with 7.

Example 2:

Input: 2, 4, 8, 3 and $n=1$

Output: 3, 4, 8, 2

Explanation: The 1st element from the start is 2 and the 1st element from the end is 3 so 2 is swapped with 3. 5. Insert a node at head and tail in $O(1)$ time

6. Create a suitable main to test the above functions.

5. Remove Nodes

remove all nodes from the double link list which contain elements whose digit sum is even and find the sum and product of the remaining element

input: 9-> 11-> 34-> 6 -> 13 -> 21

output: 9->34->21

sum : 64

product : 6426

Question 2:

Children often play a counting-out game to randomly select one person from the group by singing a rhyme. The purpose is to select one person, either as a straightforward winner, or as someone who is eliminated. Josephus problem is related to this concept. In this problem, people are standing in one circle waiting to be executed.

More Explanation: N people, numbered 1 to N, are sitting in a circle. Starting at person 1, a token is passed. After M passes, the person holding the token is eliminated, the circle closes together, and

the game continues with the person who was sitting after the eliminated person picking up the token. The last remaining person wins. See the following examples:

- If $M = 0$ and $N = 5$, players are eliminated in order 1, 2, 3, 4, and player 5 wins.
- If $M = 1$ and $N = 5$, the order of elimination is 2, 4, 1, 5, and player 3 wins.
- If $M = 2$ and $N = 5$, the order of elimination is 3, 1, 5, 2, and player 4 wins.

You are required to write a program to solve the Josephus problem for general values of M and N , according to the following specifications:

```
Class JosephusList;
```

```
Class PersonNode
```

```
{
    friend class JosephusList;
    private:
        int id;
        PersonNode*next;
};
class JosephusList
{
    private:
        PersonNode* head;
    public: ...
};
```

The JosephusList class will have the following public member functions: JosephusList (int N) This is the constructor of JosephusList class. This constructor will create a circular singly linked list of N person-nodes containing values from 1 to N . For example, if at declaration time the user specifies N to be 5, this constructor should create a circular linked list containing these 5 values {1 2 3 4 5}.

~JosephusList () : Destructor

int getWinner (int M) : This function will take M as an argument, and will return the number of the winning person determined through the process described above. During its execution, this function will display the numbers corresponding to the persons which are being eliminated. Make sure that the nodes corresponding to the eliminated persons are deleted from the list correctly. When this function completes its execution, there should be only one node (the winner) left in the list.

Main should be like this:

```
int main()
{
```

```
    JosephusList j(5);  
    Cout<<j.getWinner(3);  
    j.display();  
    system("pause");  
return 0;  
}
```

Question 3:

Given a sorted doubly linked list of distinct nodes(no two nodes have the same data) and a value x. The task is to count the triplets in the list that produce up to a given value x.

Input: list = 1->2->4->5->6->8->9, x = 8

Output: 1

Triplet is (1, 2, 4)