Personal Information

Name: Khan Muhammad

CMS ID: 023-22-0199

**Assignment 3**

Develop a linked list (Doubly) with the following methods

1.       Add an element at the start of the linked list.

2.       Remove an element at the start of the linked list.

3.       Search for an element in the linked list.

4.       Update an element in the linked list.

5.       Check linked list is empty ?

**Source Code:**

class Node

{

    int data;

    Node next;

    Node prev;

    Node(int data)

    {

        this.data = data;

        this.next = null;

        this.prev = null;

    }

}

class LinkedList

{

    Node head;

    Node tail;

    LinkedList()

    {

        this.head = null;

        this.tail = null;

    }

    LinkedList(int data)

    {

        Node newNode = new Node(data);

        head = newNode;

        tail = newNode;

    }

    void pushFront(int data)        // Push Front : Adds Element at the start.

    {

        Node newNode = new Node(data);

        if(isEmpty())

        {

            this.head = newNode;

            this.tail = newNode;

        }

        else

        {

            head.prev = newNode;

            newNode.next = head;

            head = newNode;

        }

    }

    int popFront()                 // Pop Front : Removes an element from the start.

    {

        if(isEmpty())

        {

            System.out.println("No Element Left To Pop!");

            return 0;

        }

        int data = head.data;

        head.next.prev = null;

        head = head.next;

        return data;

    }

    boolean search(int data)          // Search : search for an element

    {

        Node temp = head;

        while(temp!=null)

        {

            if(data==temp.data)

            {

                return true;

            }

            temp = temp.next;

        }

        return false;

    }

    void updateNode(int oldData, int newData)       // Update Node : updates nodes data

    {

        if(search(oldData))

        {

            Node temp = head;

            while(temp!=null)

            {

                if(oldData==temp.data)

                {

                    temp.data = newData;

                }

                temp = temp.next;

            }

        }

        else

        {

            System.out.println(oldData+" NOT FOUND!");

        }

    }

    boolean isEmpty()               // Is Empty

    {

        if(this.head==null)

            return true;

        else

            return false;

    }

    void printList()                // Print List

    {

        Node temp = head;

        System.out.print("{");

        while(temp!=null)

        {

            System.out.print(temp.data+", ");

            temp = temp.next;

        }

        System.out.println("\b\b}");

    }

}

class Assignment3

{

    public static void main(String [] args)

    {

        LinkedList l1 = new LinkedList(12);

        l1.pushFront(11);

        l1.pushFront(10);

        System.out.print("\nList: ");

        l1.printList();

        System.out.print("\n1) Adding '5' at the start: ");

        l1.pushFront(5);

        l1.printList();

        System.out.print("2) Removing the element at start: ");

        int popValue = l1.popFront();

        l1.printList();

        System.out.print("\t\t\t'"+popValue+"' Was Popped!");

        System.out.print("\n\n3) i.  Search 23: "+l1.search(23));

        System.out.print("\n3) ii. Search 11: "+l1.search(11));

        System.out.print("\n\n4) i.  Update Element 11 with 7: ");

        l1.updateNode(11, 7);

        l1.printList();

        System.out.print("4) ii. Update Element 4 with 14: ");

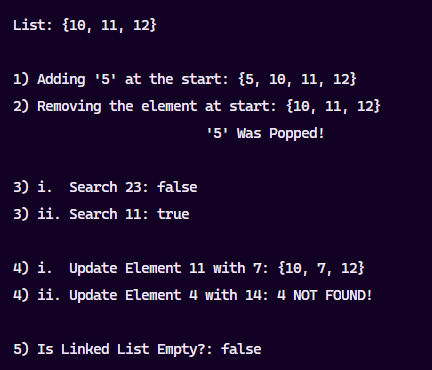
        l1.updateNode(4, 14);

        System.out.print("\n5) Is Linked List Empty?: "+l1.isEmpty()+"\n\n");

    }

}

**Result:**

****