Bahria University,

Karachi Campus

A picture containing text, room

Description automatically generated

LAB EXPERIMENT NO.

**04**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
|  | **Circular and doubly linked list** |
| 1 | Write a program to create a linked list and pertform  \*traversing  \* Insertion  \*deletion |

Submitted On:

(Date: DD/MM/YY)

**TASK #04:** Write a program to create a linked list and perform

\*traversing

\* Insertion

\*deletion

**SOLUTION:**

public class Node

{

public Node next;

public Node prev;

public int data;

public Node(int d)

{

data = d;

}

}

public class Dll

{

public Node Head;

public Dll()

{

Head = null;

}

public void print(Node node)

{

Node last = null;

Console.WriteLine("PRENTING FORWARD DIRECTION:");

while (node != null)

{

Console.WriteLine(node.data + " ");

last = node;

node = node.next;

}

Console.WriteLine("PRENTING REVERSE DIRECTION:");

while (last != null)

{

Console.WriteLine(last.data + " ");

last = last.prev;

}

}

public void insert\_in\_front(int newElement)

{

Node NEWnODE = new Node(newElement);

NEWnODE.data = newElement;

NEWnODE.next = null;

NEWnODE.prev = null;

if (Head == null)

{

Head = NEWnODE;

}

else

{

Head.prev = NEWnODE;

NEWnODE.next = Head;

Head = NEWnODE;

}

}

public void insert\_At\_End(int element)

{

Node newnode = new Node(element);

newnode.data = element;

newnode.next = null;

Node last = Head;

if (Head == null)

{

newnode.prev = null;

Head = newnode;

return;

}

else

{

while (last.next != null)

last = last.next;

last.next = newnode;

newnode.prev = last;

}

}

public void middle(Node prev\_node ,int data) {

if (prev\_node==null)

{

Console.WriteLine("Node NOT BE NULL :");

}

Node new\_node = new Node(data);

new\_node.next = prev\_node.next;

prev\_node.next = new\_node;

new\_node.prev = prev\_node;

if (new\_node.next!=null)

{

new\_node.prev = new\_node;

}

}

public void Delete\_Element(Node value) {

if (Head==null || value==null)

{

return;

}

if (Head==value)

{

Head = value.next;

}

if (value.next!=null)

{

value.next.prev = value.prev;

}

if (value.prev != null)

{

value.prev.next = value.next;

}

} }

class Program

{

**static void Main(string[] args)**

Dll Mydll = new Dll();

Node first = new Node(5);

first.data = 5;

first.prev = null;

first.next = null;

Mydll.Head = first;

Node s = new Node(12);

s.data = 12;

s.prev = first;

s.next = null;

first.next = s;

Node t = new Node(3);

t.data = 3;

t.prev = s;

t.next = null;

s.next = t;

char ch;

do

{

Console.WriteLine("WHAT OPERATION YOU WANT TO PERFORM :");

Console.WriteLine("a)Traversing \nb)Insertion \nc)Deletion");

string option = Console.ReadLine();

Console.WriteLine();

if (option=="a")

{

Mydll.print(Mydll.Head);

}

else if (option=="b")

{

Console.WriteLine("Enter The Position Where You Want To Insert :");

Console.WriteLine("1)AT FRONT \n2)AT MIDDLE \n3)AT END");

int re = int.Parse(Console.ReadLine());

if (re==1)

{

Mydll.insert\_in\_front(44);

Mydll.print(Mydll.Head);

}

else if (re==2)

{

Mydll.middle(first, 55);

Mydll.print(Mydll.Head);

}

else if (re==3)

{

Mydll.insert\_At\_End(90);

Mydll.print(Mydll.Head);

} }

else if (option=="c")

{

Mydll.Delete\_Element(t);

Mydll.print(Mydll.Head);

}

Console.WriteLine("If You Want To Perform Again :[Y/N]");

ch= Char.Parse(Console.ReadLine());

} while (ch=='y');

Console.ReadLine();

} }}

**OUTPUT:**

**Text

Description automatically generated**