**Question1**: Write a program to create a linked list**.**

**Code:**

**LinkedList Class:**

class LinkedList

{

int length;

Node start;

public LinkedList()

{

start = new Node();

}

public LinkedList(Node obj)

{

start = obj;

}

public bool Underflow()

{

if (start.next != null)

{

return false;

}

else

{

return true;

}

}

public bool InsertAtBeginning(int val)

{

Node n = new Node(val);

if (!Underflow())

{

n.next=start.next;

start.next = n;

return true;

}

else

{

start.next = n;

return true;

}

}

public bool InsertAtEnd(int val)

{

Node n = new Node(val);

if (!Underflow())

{

Node temp = start.next;

while (temp.next != null)

{

temp = temp.next;

}

temp.next = n;

return true;

}

else

{

start.next = n;

return true;

}

}

public bool InsertAfter(int val, int newval)

{

Node n = new Node(newval);

if (!Underflow())

{

Node temp = start.next;

while (temp.data == newval || temp.next != null)

{

if (temp.data == val)

{

n.next = temp.next;

temp.next = n;

return true;

}

else

{

temp = temp.next;

}

}

if (InsertAtEnd(newval))

{

Console.WriteLine("Value not found");

Console.WriteLine("New value inserted at the end");

return true;

}

return false;

}

else

{

if (InsertAtEnd(newval))

{

Console.WriteLine("Value not found");

Console.WriteLine("New value inserted at the end");

return true;

}

return false;

}

}

public bool RemoveFirst()

{

if (!Underflow())

{

if (start.next.next != null)

{

start.next = start.next.next;

return true;

}

else

{

start.next = null;

return true;

}

}

return false;

}

public bool RemoveLast()

{

if (!Underflow())

{

if (start.next.next == null)

{

start.next = null;

return true;

}

else {

Node temp = start.next;

while (temp.next.next != null)

{

temp = temp.next;

}

temp.next = null;

return true;

}

}

return false;

}

public void Display()

{

if (!Underflow())

{

Node temp = start.next;

Console.WriteLine("Your list");

while (temp.next != null)

{

Console.Write(temp.data+" ");

temp = temp.next;

}

Console.Write(temp.data);

Console.WriteLine();

}

else

{

Console.WriteLine("List empty");

}

}

public bool RemoveAfter(int val)

{

if (!Underflow())

{

Node temp = start.next;

while(temp.next!=null || temp.data == val)

{

if (temp.data == val)

{

if (temp.next.next != null)

{

temp.next = temp.next.next;

return true;

}

else

{

temp.next = null;

return true;

}

}

}

}

return false;

}

}

**Node Class:**

class Node

{

internal Node next;

internal int data;

public Node()

{

next = null;

data = 0;

}

public Node(int val)

{

this.data = val;

}

public Node(Node obj)

{

this.next = obj;

}

}

**Program Class:**

class Program

{

static void Main(string[] args)

{

LinkedList obj = new LinkedList();

while (true) {

Console.WriteLine("What action you want to perform?");

Console.WriteLine("1-Add item in the beginning\n2-Add at last\n3-Add after any number");

Console.WriteLine("4-Remove first\n5-Remove last \n6-Remove after\n0-Exit");

int choice = Convert.ToInt32(Console.ReadLine());

if (choice == 1)

{

Console.WriteLine("Enter new value");

int value = Convert.ToInt32(Console.ReadLine());

if (obj.InsertAtBeginning(value))

{

Console.WriteLine("Value added");

obj.Display();

}

}

else if (choice == 2)

{

Console.WriteLine("Enter new value");

int value = Convert.ToInt32(Console.ReadLine());

if (obj.InsertAtEnd(value))

{

Console.WriteLine("Value added");

obj.Display();

}

}

else if (choice == 3)

{

Console.WriteLine("Enter value after which you want to add number");

int value = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter new value");

int newvalue = Convert.ToInt32(Console.ReadLine());

if (obj.InsertAfter(value, newvalue))

{

Console.WriteLine("Value added");

obj.Display();

}

}

else if (choice == 4)

{

if (obj.RemoveFirst())

{

Console.WriteLine("Value removed");

obj.Display();

}

else

{

Console.WriteLine("an error occured");

}

}

else if (choice == 5)

{

if (obj.RemoveLast())

{

Console.WriteLine("Value removed");

obj.Display();

}

else

{

Console.WriteLine("An error occured.");

}

}

else if (choice == 6)

{

Console.WriteLine("Enter value after which you want to remove number");

int value = Convert.ToInt32(Console.ReadLine());

if (obj.RemoveAfter(value))

{

Console.WriteLine("Value removed");

obj.Display();

}

else

{

Console.WriteLine("Value not found");

obj.Display();

}

}

else if (choice == 0)

{

break;

}

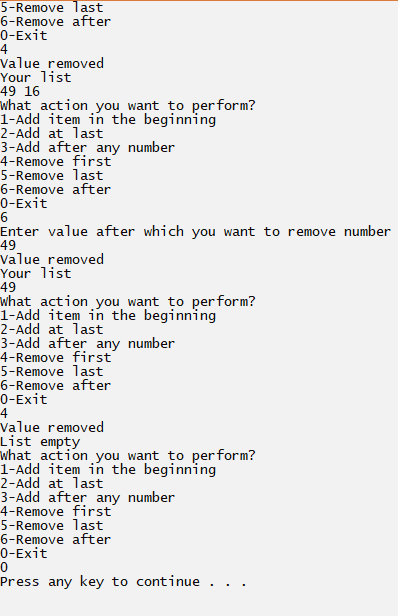
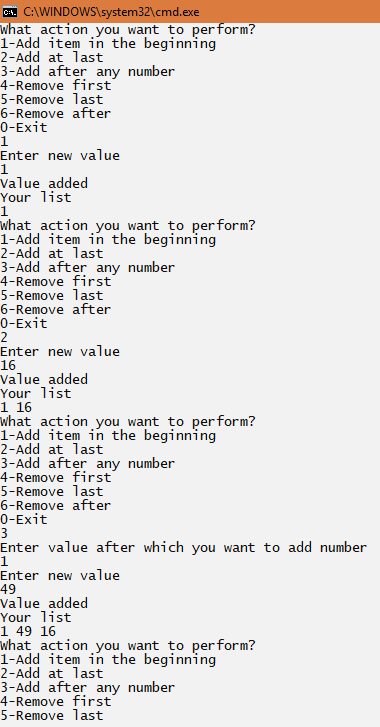
else{

Console.WriteLine("Kindly select correct option");

} }

}}

**Output:**



**Question2:** Write a program to implement dynamic stacks by using linked list.

**Code:**

**LinkedList/Stack class:**

class LinkedList

{

Node start;

public LinkedList()

{

start = new Node();

}

public LinkedList(Node obj)

{

start = obj;

}

public bool Underflow()

{

if (start.next != null)

{

return false;

}

else

{

return true;

}

}

public bool Push(int val)

{

Node n = new Node(val);

if (!Underflow())

{

n.next=start.next;

start.next = n;

return true;

}

else

{

start.next = n;

return true;

}

}

public int Peek()

{

if (!Underflow())

{

int val;

val = start.next.data;

return val;

}

return -1;

}

public int Pop()

{

if (!Underflow())

{

int val;

if (start.next.next != null)

{

val = start.next.data;

start.next = start.next.next;

return val;

}

else

{

val = start.next.data;

start.next = null;

return val;

}

}

return -1;

}

public void Display()

{

if (!Underflow())

{

Node temp = start.next;

Console.WriteLine("Your list");

while (temp.next != null)

{

Console.Write(temp.data+" ");

temp = temp.next;

}

Console.Write(temp.data);

Console.WriteLine();

}

else

{

Console.WriteLine("List empty");

}

}

}

**Program Class:**

static void Main(string[] args)

{

LinkedList obj = new LinkedList();

while (true) {

Console.WriteLine("What action you want to perform?");

Console.WriteLine("1-Push\n2-Pop\n3-Peek\n4-Exit");

int choice = Convert.ToInt32(Console.ReadLine());

if (choice == 1)

{

Console.WriteLine("Enter new value");

int value = Convert.ToInt32(Console.ReadLine());

if (obj.Push(value))

{

Console.WriteLine("Value pushed");

obj.Display();

}

}

else if (choice == 2)

{

int val= obj.Pop();

if (val!=-1)

{

Console.WriteLine("Value popped");

Console.WriteLine("Popped value = "+val);

obj.Display();

}

else

{

Console.WriteLine("Stack underflow");

}

}

else if (choice == 3)

{

int val = obj.Peek();

if (val != -1)

{

Console.WriteLine("Peek value = " + val);

}

else

{

Console.WriteLine("Stack underflow");

}

}

else if (choice == 0)

{

break;

}

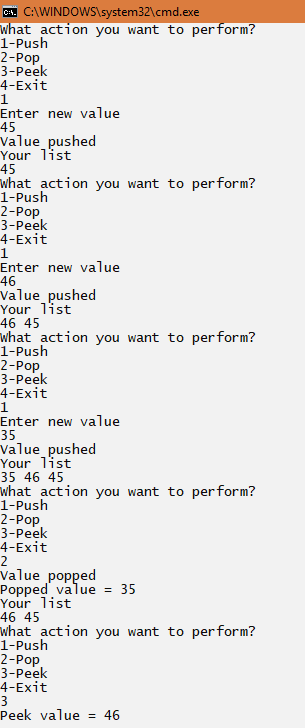
else{

Console.WriteLine("Kindly select correct option");

} }

}}

**Output:**



**Question3:** Write a program to implement dynamic queue by using linked list.

**Code:**

**Linkedlist/Queue class:**

class LinkedList

{

Node start;

public LinkedList()

{

start = new Node();

}

public LinkedList(Node obj)

{

start = obj;

}

public bool Underflow()

{

if (start.next != null)

{

return false;

}

else

{

return true;

}

}

public bool Enqueue(int val)

{

Node n = new Node(val);

if (!Underflow())

{

n.next=start.next;

start.next = n;

return true;

}

else

{

start.next = n;

return true;

}

}

public int Top()

{

if (!Underflow())

{

int tem;

if (start.next.next == null)

{

tem = start.next.data;

return tem;

}

else

{

Node temp = start.next;

while (temp.next.next != null)

{

temp = temp.next;

}

tem = temp.next.data;

return tem;

}

}

return -1;

}

public int Dequeue()

{

if (!Underflow())

{

int tem;

if (start.next.next == null)

{

tem = start.next.data;

start.next = null;

return tem;

}

else {

Node temp = start.next;

while (temp.next.next != null)

{

temp = temp.next;

}

tem = temp.next.data;

temp.next = null;

return tem;

}

}

return -1;

}

public void Display()

{

if (!Underflow())

{

Node temp = start.next;

Console.WriteLine("Your list");

while (temp.next != null)

{

Console.Write(temp.data+" ");

temp = temp.next;

}

Console.Write(temp.data);

Console.WriteLine();

}

else

{

Console.WriteLine("List empty");

}

}

}

**Program Class:**

class Program

{

static void Main(string[] args)

{

LinkedList obj = new LinkedList();

while (true) {

Console.WriteLine("What action you want to perform?");

Console.WriteLine("1-Enqueue\n2-Dequeue\n3-Top\n4-Exit");

int choice = Convert.ToInt32(Console.ReadLine());

if (choice == 1)

{

Console.WriteLine("Enter new value");

int value = Convert.ToInt32(Console.ReadLine());

if (obj.Enqueue(value))

{

Console.WriteLine("Value enqueued");

obj.Display();

}

}

else if (choice == 2)

{

int val= obj.Dequeue();

if (val!=-1)

{

Console.WriteLine("Value dequeued");

Console.WriteLine("Dequeued value = "+val);

obj.Display();

}

else

{

Console.WriteLine("Queue underflow");

}

}

else if (choice == 3)

{

int val = obj.Top();

if (val != -1)

{

Console.WriteLine("Top value = " + val);

}

else

{

Console.WriteLine("Queue underflow");

}

}

else if (choice == 0)

{

break;

}

else{

Console.WriteLine("Kindly select correct option");

} }

}

}

**Output:**

