Task No 01:

Write a program to create a linked list and perform:

1. Traversing
2. Insertion
3. Deletion

Solution:

1. Traversing:

Main:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

class Program

{

static void Main(string[] args)

{

LinkedList llist = new LinkedList();

llist.head = new Node(1);

Node second = new Node(2);

Node third = new Node(3);

llist.head.next = second;

second.next = third;

llist.printList();

}

}

}

LinkedList Class:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class LinkedList

{

public Node head;

public void printList()

{

Node n = head;

while (n != null)

{

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

n = n.next;

}

}

}

}

Node Class:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class Node

{

public int data;

public Node next;

public Node(int d)

{

data = d;

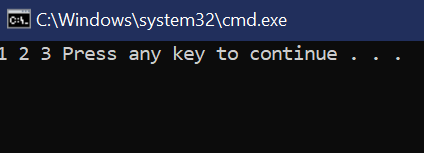
next = null;

}

}

}

Output:



1. Insertion:

At the front of the linked list:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class LinkedList

{

public Node head;

public void push(int new\_data)

{

Node new\_node = new Node(new\_data);

new\_node.next = head;

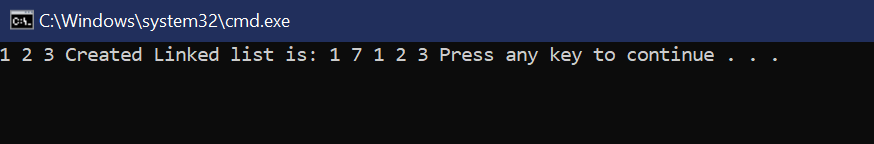
head = new\_node;

}

}

}

Output:



After a given node:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class LinkedList

{

public Node head;

public void insertAfter(Node prev\_node, int new\_data)

{

if (prev\_node == null)

{

Console.WriteLine("The given previous" + " node cannot be null");

return;

}

Node new\_node = new Node(new\_data);

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

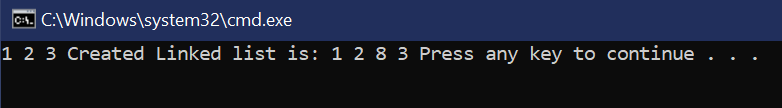
prev\_node.next = new\_node;

}

}

}

Output:



At the end of the linked list:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class LinkedList

{

public Node head;

public void append(int new\_data)

{

Node new\_node = new Node(new\_data);

if (head == null)

{

head = new Node(new\_data);

return;

}

new\_node.next = null;

Node last = head;

while (last.next != null)

last = last.next;

last.next = new\_node;

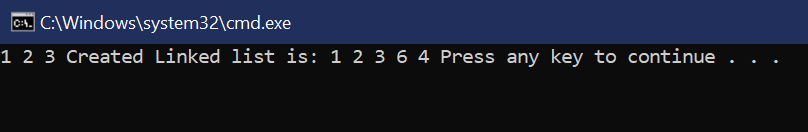
return;

}

}

}

Output:



Main:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

class Program

{

static void Main(string[] args)

{

LinkedList llist = new LinkedList();

llist.head = new Node(1);

Node second = new Node(2);

Node third = new Node(3);

llist.head.next = second;

second.next = third;

GFG list = new GFG();

llist.append(6);

llist.push(7);

llist.push(1);

llist.append(4);

llist.insertAfter(llist.head.next, 8);

Console.Write("Created Linked list is: ");

llist.printList();

}

}

}

1. Deletion:

Main:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

class Program

{

static void Main(string[] args)

{

LinkedList llist = new LinkedList();

llist.head = new Node(1);

Node second = new Node(2);

Node third = new Node(3);

llist.head.next = second;

second.next = third;

llist.deleteNode(1);

}

}

}

LinkedList Class:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab03

{

public class LinkedList

{

public Node head;

public void deleteNode(int key)

{

Node temp = head, prev = null;

if (temp != null && temp.data == key)

{

head = temp.next;

return;

}

while (temp != null && temp.data != key)

{

prev = temp;

temp = temp.next;

}

if (temp == null)

return;

prev.next = temp.next;

}

}

}

Output:

