**كود التعامل مع nodes**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace linklist
{
   class Program
   {
      class Node
      {
         public int data;
         public Node next;
         public Node(int d)
         {
            data = d;
            next = null;
         }
      }
         static void Main(string[] args)
         {
           Node one=new Node(1);
           Node two = new Node(2);
            Node three = new Node(3);
         Console.WriteLine(one.data);
         Console.WriteLine(two.data);
         Console.WriteLine(three.data);
```

```
        Console.WriteLine("linked between nodes");
        one.next = two;
        two.next = three;
        Console.WriteLine(one.next.data);
        Console.WriteLine(one.next.next.data);
        Console.WriteLine(one.next.next.next.data);

    }
    }
  }
```

**كود اضافة عنصر نهاية linked list وبداية linked list وفي وسط linked list والبحث عن node وطباعة node وحذف node**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace linklist
{
  class Program
  {
    class Node
    {
      public int data;
      public Node next;
      public Node(int d)
      {
```

```
        data = d;
        next = null;

    }
}
class linkedlist
{
    Node head;
    Node tail;
    public linkedlist()
    {
        head = tail = null;
    }
    public void Insertend(int new_data)
    {
        Node new_node = new Node(new_data);

        if (head == null)
        {
            head = new_node;
            tail = new_node;
        }
        else
        {
            tail.next = new_node;
            tail = new_node;

        }
    }
```

```
        public void Insertfirst(int new_data)
        {
            Node new_node = new Node(new_data);

            if (head == null)
            {
                head = new_node;
                tail = new_node;
            }
            else
            {
                new_node.next = head;
                head = new_node;

            }
        }
اضافة في الوسط          public void insert(int value, int pos) //
        {
            Node current = head;
            for (int i = 0; i<pos - 1; i++)
            {
                current = current.next;
                if (current == null)
                {
                    Console.WriteLine("out range");
                    return;
                }
    }
    Node new_node = new Node(value);
```

```
    new_node.next = current.next;
            current.next = new_node;
          }
public void search(int value)
{
    Node current = head;
    int pos = 0;
    while (current != null)
    {
        if (current.data == value)
        {
            Console.WriteLine($"item {value}found at {pos}");
            return;
        }
        current = current.next;
        pos++;

    }
    if (current == null)
    {
        Console.WriteLine("not found");
    }
}
public void delete(int value)
{
    if (head == null)
    {
        Console.WriteLine(" linked list is empty");
        return;
```

```
    }
    if (head.data == value)
    {
        head = head.next;
        return;
    }
    Node previous = head;
    Node current = head.next;
    while (current != null)
    {
        if (current.data == value)
        {
            previous.next = current.next;
            if (previous.next == null)
            {
                tail = previous;
            }
            return;
        }
        previous = previous.next;
        current = current.next;
    }
    current = head;
    if (current == null)
    { Console.WriteLine("not found");

    }
   }
public void PrintLinkedList()
```

```
{
    Node current = head;
    if (head == null)
    { Console.WriteLine("linked list is empty"); }
    else
    {
        while (current != null)
        {
            Console.WriteLine(current.data);
            current = current.next;
        }
    }

}

static void Main(string[] args)
{
    linkedlist ll = new linkedlist();
    ll.Insertfirst(78);
    ll.Insertfirst(100);
    ll.Insertfirst(10);
    ll.Insertend(15);
    ll.insert(500, 3);
    ll.PrintLinkedList();
    Console.WriteLine("////////////////////////");
    ll.search(5);
    Console.WriteLine("after delete");
    ll.delete(500);
    ll.Insertend(444);
```

```
    ll.PrintLinkedList();


}
      }
   }
}
```