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


namespace lab2
{
    class Program
    {
        Node head;
        public class Node
        {
            public int data;
            public Node next;
            public Node(int d)
            {
                data = d;
                next = null;

            } // Constructor
        }

        public void Add(int data)
        {
            Node node = new Node(data);

            if (head == null)
            {
                head = new Node(data);
                return;
            }
            node.next = null;
            Node last = head;
            while (last.next != null)
                last = last.next;
            last.next = node;
            return;
        }

        public void printList()
        {
            Node n = head;
            while (n != null)
            {
                Console.Write(n.data + " ");
                n = n.next;
            }
            Console.WriteLine();
        }
```

```
public void getFirst()
{
   Node n = head;
   Console.Write(n.data);
   Console.WriteLine();
}

public void getLast()
{
   Node n = head;
   while ( n != null)
   {
      n = n.next;
      if (n.next == null)
      {
         Console.Write(n.data);
         break;
      }

   }
   Console.WriteLine();

}

public void GetNext()
{
   Console.Write("Input position to get next item: ");
   int x = Convert.ToInt32(Console.ReadLine());
   Node n = head;
   int count =1;
   while (n!= null)
   {
      if (count == x)
      {
         Console.Write("Get Next: {0}", n.next.data);
         Console.WriteLine();
      }
      n = n.next;
      count++;
   }
   Console.WriteLine();
}

public void GetPrevious()
{
   Console.Write("Input position to get previous item: ");
   int x = Convert.ToInt32(Console.ReadLine());
   Node n = head;
   int count =1;
   while (n!= null)
   {
      if (count < x)
```

```
        {
            Console.Write("Get Previous: {0}",n.next.data);
            break;
        }
        n = n.next;
        count++;
    }
    Console.WriteLine();
}

public void get()
{
    Console.WriteLine();
    Console.Write("Input position to get item: ");
    int x = Convert.ToInt32(Console.ReadLine());
    Node n = head;
    int count =1;
    while (n!= null)
    {
        if (count == x)
        {
            Console.Write("Get item at position {0}: {1}",x, n.data);
            break;
        }
        n = n.next;
        count++;
    }
    Console.WriteLine();
}

public void set()
{
    Console.WriteLine();
    Console.Write("Input position to set item: ");
    int x = Convert.ToInt32(Console.ReadLine());
    Console.Write("Input value to set: ");
    int vl = Convert.ToInt32(Console.ReadLine());
    Node n = head;
    int count =1;
    while (n!= null)
    {
        if (count == x)
        {
            n.data=vl;
            break;
        }
        n = n.next;
        count++;
    }
}

public void Insert()
```

```
{
   Console.WriteLine();
   Console.Write("Input position to insert item: ");
   int x = Convert.ToInt32(Console.ReadLine());
   Console.Write("Input value to insert: ");
   int vl = Convert.ToInt32(Console.ReadLine());
   Node value = new Node(vl);
   Node n = head;
   int count =1;

   while (n != null)
   {
      if (count == x-1)
      {
         value.next = n.next;
         n = value;
         break;
      }
      n = n.next;
      count++;
   }
}

static void Main(string[] args)
{
   Program lList = new Program();
   bool key =true;
   int n=1;
   while (key)
   {

      Console.Write("Input value {0} for Linked List: ",n);
      int a = Convert.ToInt32(Console.ReadLine());
      if (a == 0)
      {
         key =false;
         break;
      }
      lList.Add(a);
      n++;

   }
   Console.WriteLine();
   Console.Write("Linked list: ");
   lList.printList();
   Console.WriteLine();
   Console.Write("Get First: ");
   lList.getFirst();
   Console.Write("Get Last: ");
   lList.getLast();
   lList.GetNext();
   lList.GetPrevious();
```

```
            lList.get();
            lList.set();
            Console.Write("Linked List after set data: ");
            lList.printList();
            lList.Insert();
            Console.Write("Linked List after insert: ");
            lList.printList();

        }
    }
}
```