***Search in a Linked List (Iterative and Recursive)***

Write a function that searches a given key ‘x’ in a given singly linked list.

**Iterative Solution:**

C++Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class Node{

int data;

Node next;

Node(int x){

data=x;

next=null;

}

}

class Test {

public static void main(String args[])

{

Node head=new Node(10);

head.next=new Node(20);

head.next.next=new Node(30);

printlist(head);

System.out.println("Position of element in Linked List: "+search(head,20));

}

static int search(Node head, int x){

int pos=1;

Node curr=head;

while(curr!=null){

if(curr.data==x)

return pos;

else{

pos++;

curr=curr.next;

}

}

return -1;

}

public static void printlist(Node head){

Node curr=head;

while(curr!=null){

System.out.print(curr.data+" ");

curr=curr.next;

}System.out.println();

}

}

**Output**

10 20 30

Position of element in Linked List: 2

**Recursive Solution:**

C++Java

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class Node{

int data;

Node next;

Node(int x){

data=x;

next=null;

}

}

class Test {

public static void main(String args[])

{

Node head=new Node(10);

head.next=new Node(20);

head.next.next=new Node(30);

printlist(head);

System.out.println("Position of element in Linked List: "+search(head,20));

}

static int search(Node head, int x){

if(head==null)return -1;

if(head.data==x)return 1;

else{

int res=search(head.next,x);

if(res==-1)return -1;

else return res+1;

}

}

public static void printlist(Node head){

Node curr=head;

while(curr!=null){

System.out.print(curr.data+" ");

curr=curr.next;

}System.out.println();

}

}

**Output**

10 20 30

Position of element in Linked List: 2