**Question No:05**

#include<iostream>

using namespace std;

class Node;

class Node{

public:

int data;

Node \*next;

Node(){

data = 0;

next = NULL;

}

Node(int d){

this->data = d;

this->next = NULL;

}

};

class LinkedList{

Node \*head;

public:

LinkedList(){

this->head = NULL;

}

void insertAtStart(int d){

Node \*newNode = new Node(d);

newNode->next = head;

head =newNode;

}

void insertAtLast(int d){

Node \*newNode = new Node(d);

Node \* temp = head;

while(temp->next!= NULL){

temp = temp->next;

}

temp->next = newNode;

}

void insertAtAnyPosition(int d, int po){

if(po < 1){

cout<<"Invalid Positoin"<<endl;

}

else if( po == 1){

insertAtStart(d);

}

else{

Node \*newNode = new Node(d);

Node \*temp = head;

for(int i = 1 ; i<po-1 ;i++){

if(temp == NULL){

cout<<"Invalid Postion";

break;

}

temp = temp->next;

}

newNode->next = temp->next;

temp->next = newNode;

}

}

void displayLinknedList(){

Node \*temp = head;

while(temp != NULL){

cout<<temp->data;

if(temp->next != NULL){

cout<<",";

}

temp = temp->next;

}

cout<<endl;

}

void displayFirstNode(){

cout<<head->data<<endl;

}

void displaylastNode(){

Node \*temp = head;

while(temp->next != NULL){

temp = temp->next;

}

cout<<temp->data<<endl;

}

void displayCenterNode(){

Node \*temp = head;

int count = 0;

while(temp != NULL){

temp= temp->next;

count++;

}

temp = head;

for(int i = 1 ; i<count; i++){

temp = temp->next;

if(i == count/2){

cout<<temp->data<<endl;

break;

}

}

}

};

int main(){

LinkedList l;

l.insertAtStart(29);

l.insertAtStart(9);

l.insertAtLast(20);

l.insertAtLast(58);

l.insertAtAnyPosition(9,5);

l.insertAtAnyPosition(13,8);

cout<<"First Linked List Node : ";

l.displayFirstNode();

cout<<"Last LinkedList Node : ";

l.displaylastNode();

cout<<"Center linkedList : ";

l.displayCenterNode();

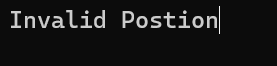
cout<<"LinkedList : ";

l.displayLinknedList();

return 0;

}

**OUTPUT**

****

**Explanation**

We implements a singly linked list with methods to insert nodes at the start, end, or a specific position. It also displays the first, last, and center nodes, along with the entire list. The main function demonstrates adding nodes and printing the list and key nodes.