**Question No:08**

#include<iostream>

using namespace std;

class Node{

public:

int data;

Node\* prev;

Node\* next;

public:

Node(int d = 0){

data = d;

prev = NULL;

next = NULL;

}

};

class DublyList{

Node\* head;

Node\* tail;

public:

DublyList(){

head = NULL;

}

void insertAtFirst(int d){

Node\* newNode = new Node(d);

if(head == NULL){

head = newNode;

}

else {

newNode->next = head;

head->prev = newNode;

head = newNode;

}

}

void insertAtLast(int d){

Node\* newNode = new Node(d);

if(head == NULL){

head = newNode;

}else{

Node\* temp = head;

while(temp->next != NULL){

temp=temp->next;

}

newNode->prev = temp;

temp->next = newNode;

}

}

void insertAtCenter(int d){

Node\* newNode = new Node(d);

if(head == NULL){

head = newNode;

}

else {

Node\* temp = head;

int count = 1;

while (temp->next != NULL)

{

temp = temp->next;

count++;

}

temp = head;

for(int i =1 ; i<=count/2;i++){

temp = temp->next;

}

newNode->next= temp->next ;

newNode->prev =temp;

temp->next->prev = newNode;

temp->next =newNode;

}

}

void insertAtSpecficPosistion(int d ,int pos){

Node\* newNode = new Node(d);

if(head == NULL){

head = newNode;

}else if(pos < 1){

cout<<"Invalid Position\n";

}else if(pos == 1){

newNode->next = head;

head->prev = newNode;

head = newNode;

}

else {

Node\* temp = head;

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

temp = temp->next;

if(temp == NULL){

break;

}

}

newNode->prev = temp;

newNode->next = temp->next;

temp->next->prev = newNode;

temp->next = newNode;

}

}

void display\_In\_Order(){

Node\* temp = head;

cout<<"Doubly List : ";

while(temp != NULL){

cout<<temp->data<<" ";

temp = temp->next;

if(temp!= NULL){

cout<< " , ";

}

}

cout<<endl;

}

void displayInreverse(){

Node\* temp = head;

cout<<"Revered Doubly List : ";

while(temp->next != NULL){

temp = temp->next;

}

while(temp != NULL){

cout<<temp->data<<" ";

if(temp->prev != NULL){

cout<<" , ";

}

temp = temp->prev;

}

cout<<endl;

}

};

int main(){

DublyList d;

d.insertAtFirst(12);

d.insertAtLast(42);

d.insertAtLast(34);

d.insertAtLast(53);

d.insertAtSpecficPosistion(27,2);

d.display\_In\_Order();

d.displayInreverse();

d.insertAtCenter(8);

cout<<"-----------------------------------------.\n";

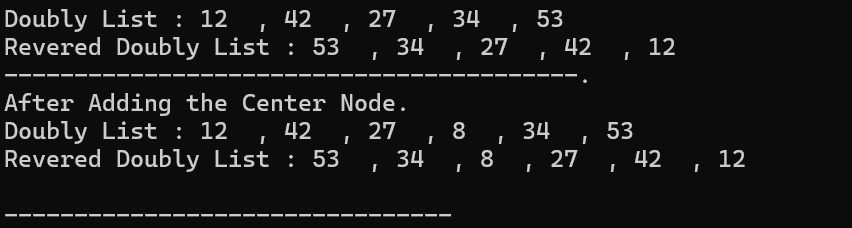
cout<<"After Adding the Center Node.\n";

d.display\_In\_Order();

d.displayInreverse();

}

**OUTPUT**

****

**Explanation**

**This code implements a** doubly Linked List **with methods to insert nodes at the start, end, center, and a specific position. It also displays the list in forward and reverse order. The main function demonstrates inserting nodes and displaying the list before and after adding a center node.**