NAME: SOBIA AHMED

SAP ID: 56424

DSA LAB 5

DOUBLY LINK LIST INSERTION AT START AND END

#include<iostream>

using namespace std;

struct Node{

int data;

Node \*next;

Node \*prev;

};

Node \*head=NULL;

void insertAtBegining(int n){

Node \*newnode=new Node;

newnode->data=n;

newnode->next=head;

newnode->prev=NULL;

if(head!=NULL)

{

head->prev=newnode;

}

head=newnode;

}

void insertAtEnd(int n)

{

Node \*newnode=new Node;

newnode->data=n;

newnode->next=NULL;

newnode->prev=NULL;

if(head==NULL)

{

head=newnode;

return;

}

Node \*temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=newnode;

newnode->prev=temp;

}

void display(){

Node \*temp=head;

while(temp!=NULL){

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

temp=temp->next;

}

}

int main(){

cout<<"insertion at start"<<endl;

insertAtBegining(1);

insertAtBegining(12);

insertAtBegining(14);

insertAtBegining(15);

display();

cout<<endl;

cout<<"insertion at end"<<endl;

insertAtEnd(8);

insertAtEnd(81);

insertAtEnd(23);

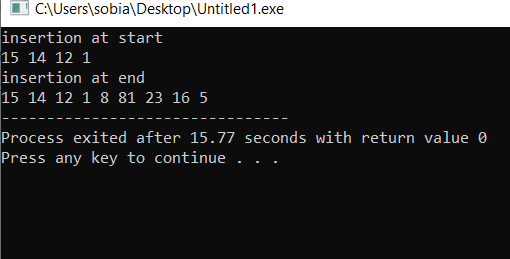
insertAtEnd(16);

insertAtEnd(5);

display();

return 0;

}



DELETION DOUBLY AT START AND END

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

Node\* prev;

};

Node\* head = NULL;

void insertAtBegining(int n) {

Node\* newnode = new Node;

newnode->data = n;

newnode->next = head;

newnode->prev = NULL;

if (head != NULL) {

head->prev = newnode;

}

head = newnode;

}

void deleteAtBegining() {

if (head == NULL) {

cout << "List is empty." << endl;

return;

}

Node\* temp = head;

head = head->next;

if (head != NULL) {

head->prev = NULL;

}

delete temp;

}

void deleteAtEnd() {

if (head == NULL) {

cout << "List is empty." << endl;

return;

}

Node \*temp=head;

while(temp->next!=NULL)

{

temp=temp->next;

}

if(temp->prev!=NULL)

{

temp->prev->next=NULL;

return;

}

head=NULL;

delete temp;

}

void display() {

cout << "Data elements in the linked list are: ";

Node\* temp = head;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

int main() {

cout<<"insertion at start"<<endl;

insertAtBegining(1);

insertAtBegining(12);

insertAtBegining(14);

insertAtBegining(15);

display();

deleteAtBegining();

cout<<"liste deleting first node"<<endl;

display();

cout<<endl;

deleteAtEnd();

cout<<"liste deleting first node"<<endl;

display();

return 0;

}

