**Name:**

Abishy Sawan

**Registration no:**

Fa21-bse-044

**Submitted To:**

Madam Yasmeen Jana

**Section:**

**A (Repeater)**

**LAB MID**

**QNO.2 : Implement stack using Array?**

#include <iostream>

using namespace std;

#define k 12

class stk{

private:

int stack[k];

int top;

public:

stk(){

top=-1;

}

void push(int w){

if(top==k-1){

cout<<"Overflow\n";

}

top++;

stack[top]=w;

}

void pop(){

if(top==-1){

cout<<"Underflow\n";

}

top--;

}

int peek(){

if(top==-1){

cout<<"Underflow\n";

}

return stack[top];

}

void display(){

for(int i=top; i>=0;i--){

cout<<stack[i]<<endl;

}}

};

int main(){

stk s;

char yes;

do {

int choice , x;

cout<<"Press 1 for push\n";

cout<<"Press 2 for pop\n";

cout<<"Press 3 for Display\n";

cout<<"Press 4 for Peeek\n";

cout<<"Press 5 to exit \n";

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

cin>>choice;

switch(choice){

case 1:

cout<<"Enter value to be pushed\n";

cin>>x;

s.push(x);

s.display();

break;

case 2:

s.pop();

s.display();

break;

case 3:

cout<<"Displaying Stack\n";

s.display();

break;

case 4:

cout<<"Peek Element is"<<s.peek()<<endl;

break;

default:

exit(0);

}

cout<<"Press y want to continue\n";

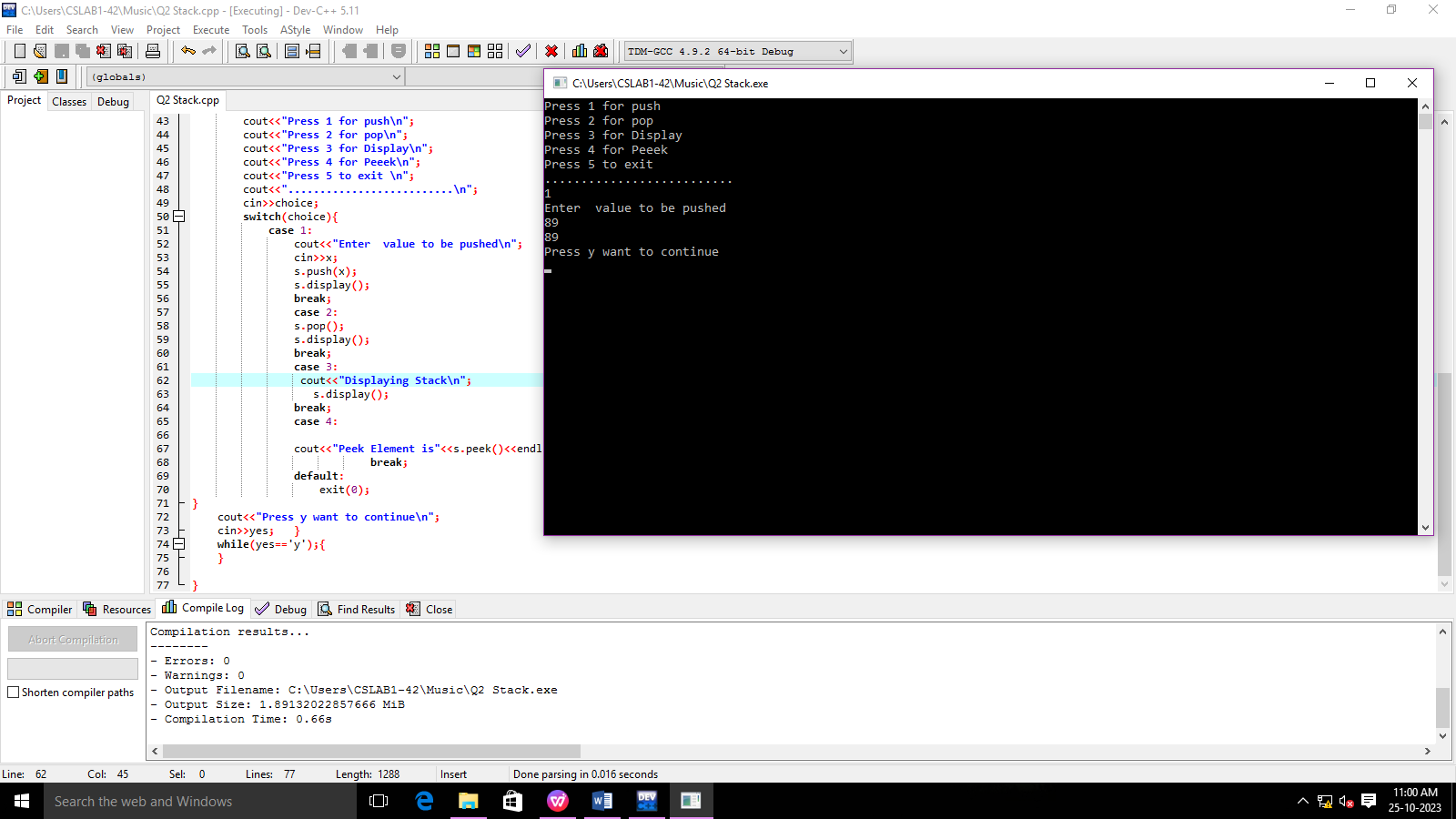
cin>>yes; }

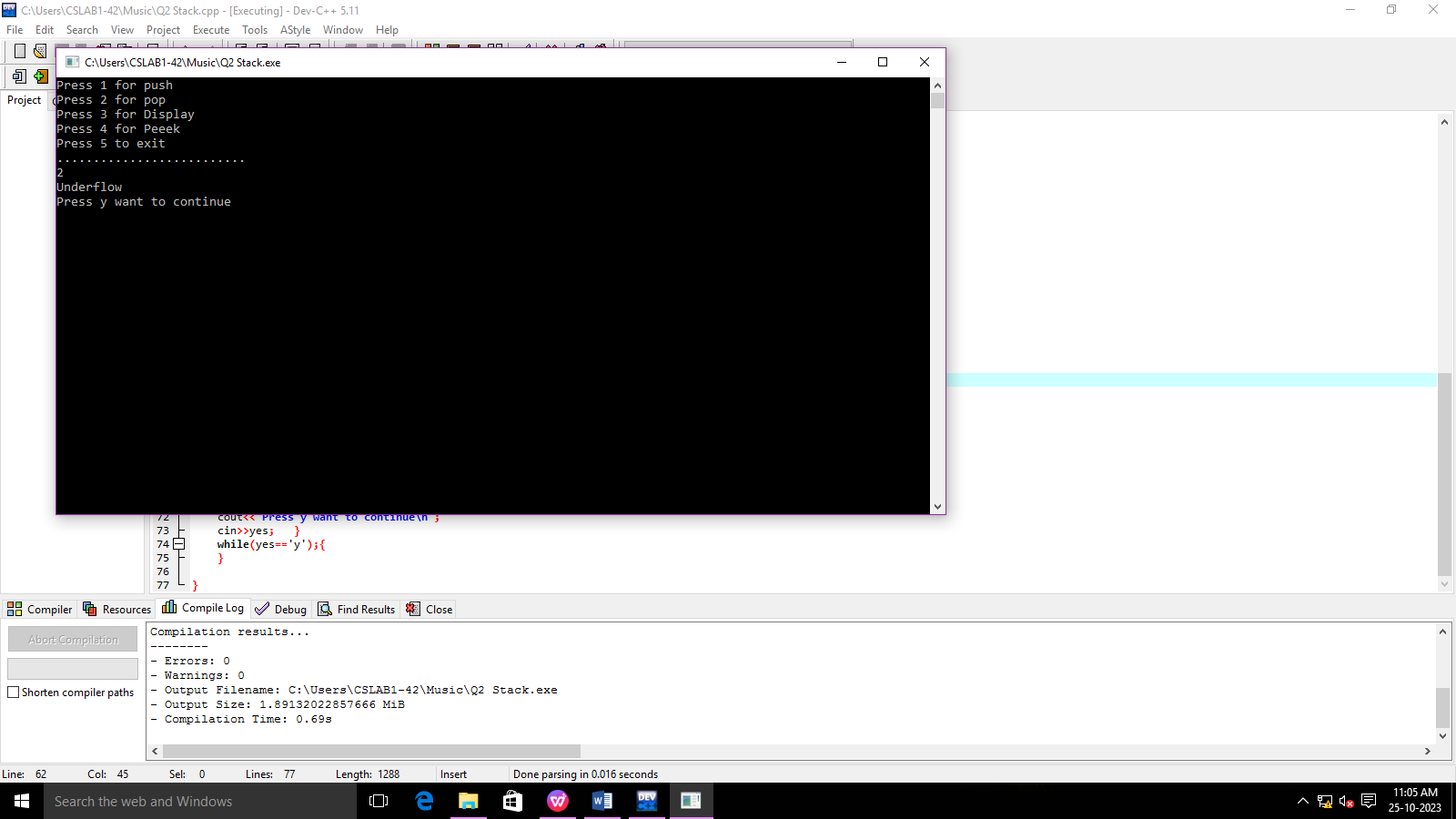
while(yes=='y');{

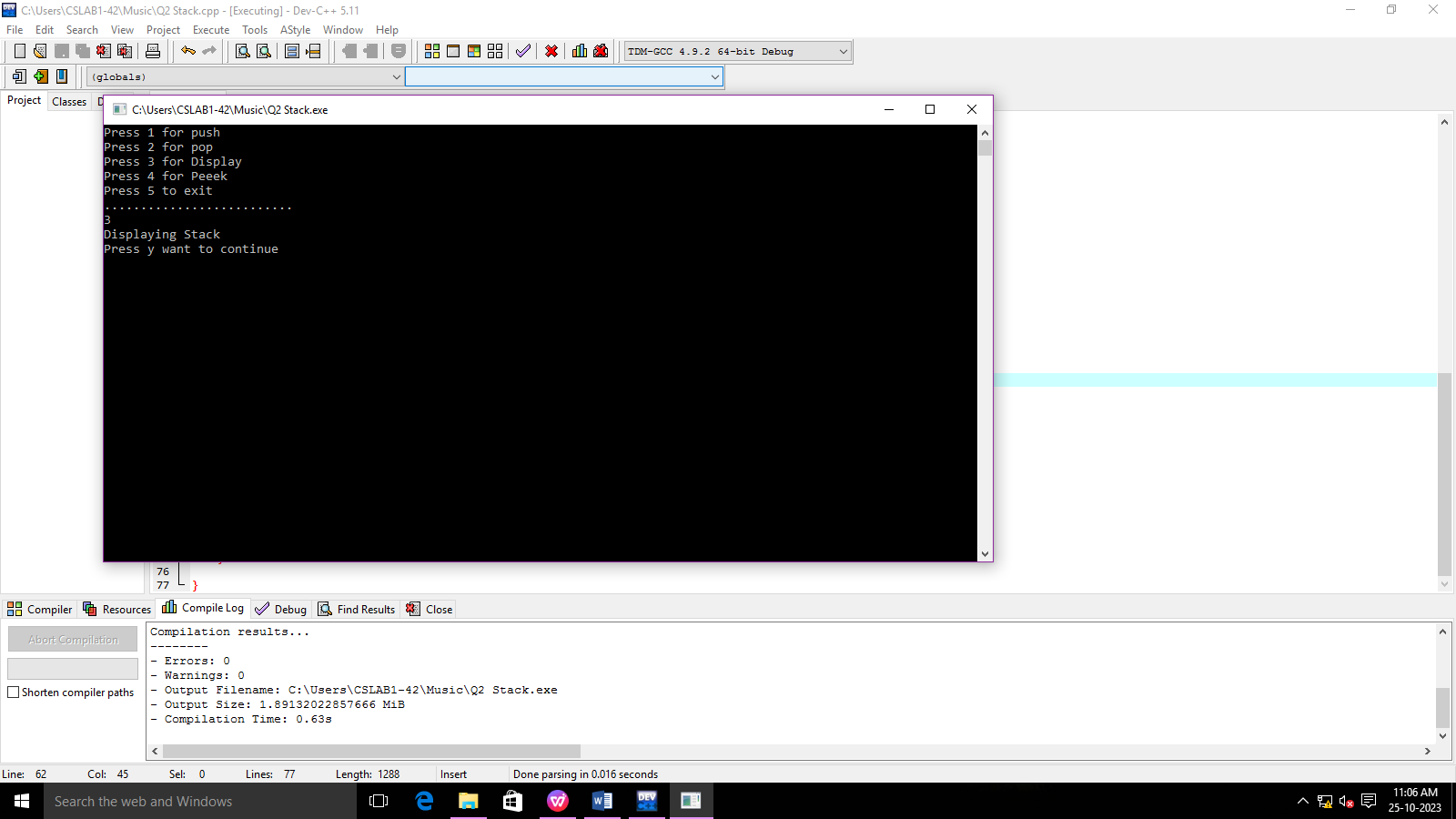
}

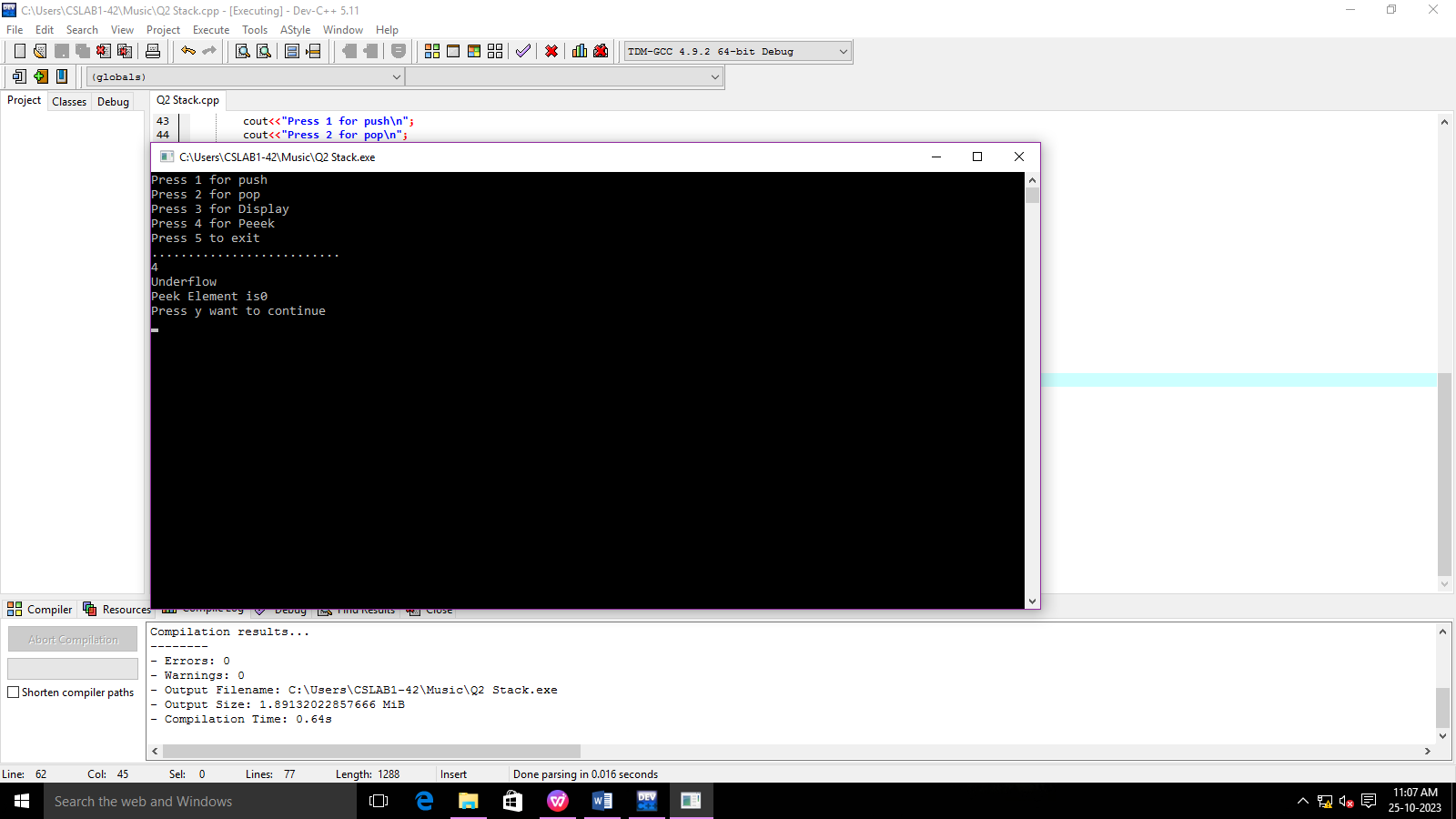
}

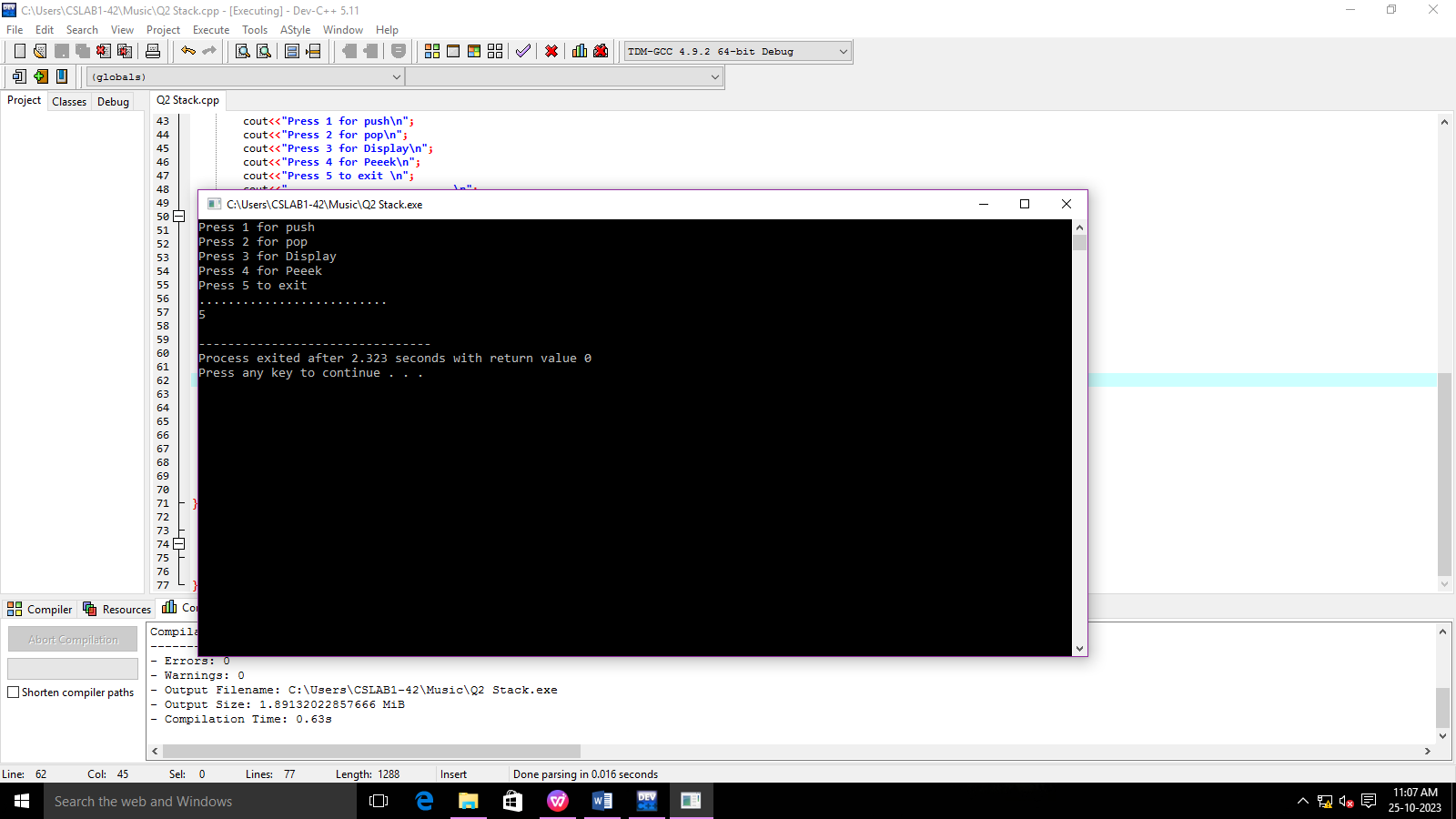
**Output:**











**QNO2: To create a single Linked list using”SLL” function and “Palindrome” function to check that Singly linked list is palindrome or not.**

#include <iostream>

#include <stack>

using namespace std;

class Node {

public:

int data;

Node\* next;

Node(int value) : data(value), next(NULL) {}

};

class LinkedList {

private:

Node\* head;

public:

LinkedList() : head(NULL) {}

void insertAtEnd(int value) {

Node\* newNode = new Node(value);

if (!head) {

head = newNode;

} else {

Node\* current = head;

while (current->next) {

current = current->next;

}

current->next = newNode;

}

}

bool isPalindrome() {

if (!head || !head->next)

return true;

stack<int> s;

Node\* slow = head;

Node\* fast = head;

while (fast && fast->next) {

s.push(slow->data);

slow = slow->next;

fast = fast->next->next;

}

if (fast)

slow = slow->next;

while (slow) {

if (s.top() != slow->data)

return false;

s.pop();

slow = slow->next;

}

return true;

}

void display() {

Node\* current = head;

while (current) {

cout << current->data;

if (current->next)

cout << " -> ";

current = current->next;

}

cout << endl;

}

};

int main() {

LinkedList list;

list.insertAtEnd(8);

list.insertAtEnd(9);

list.insertAtEnd(3);

list.insertAtEnd(14);

cout << "Linked List: ";

list.display();

if (list.isPalindrome())

cout << "The linked list is a palindrome." << endl;

else

cout << "The linked list is not a palindrome." << endl;

return 0;

}

**Output:**

