***DSA Lab Midterm***

**Submitted by:**

**Name: Afsah Ahmad**

**Reg # SP22-BCS-006(A)**

**Submitted to:**

**Ma’am Yasmeen Jana**

**Date:**

**25 October,2023**

**COMSATS University Islamabad**

**Vehari Campus**

**Q#1:**

**Single Linked List to check Palindrome**

**Code:**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

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

};

Node\* createNode(int val) {

return new Node(val);

}

Node\* SLL(int arr[], int n) {

if (n == 0) {

return NULL;

}

Node\* head = createNode(arr[0]);

Node\* current = head;

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

current->next = createNode(arr[i]);

current = current->next;

}

return head;

}

Node\* reverseList(Node\* head) {

Node\* prev = NULL;

Node\* current = head;

while (current != NULL) {

Node\* next = current->next;

current->next = prev;

prev = current;

current = next;

}

return prev;

}

bool isPalindrome(Node\* head) {

if (head == NULL || head->next == NULL) {

return true;

}

Node\* slow = head;

Node\* fast = head;

Node\* prev\_slow = head;

Node\* mid = NULL;

bool isPalin = true;

while (fast != NULL && fast->next != NULL) {

fast = fast->next->next;

prev\_slow = slow;

slow = slow->next;

}

if (fast != NULL) {

mid = slow;

slow = slow->next;

}

Node\* secondHalf = slow;

prev\_slow->next = NULL;

secondHalf = reverseList(secondHalf);

Node\* p1 = head;

Node\* p2 = secondHalf;

while (p1 != NULL && p2 !=NULL) {

if (p1->data != p2->data) {

isPalin = false;

break;

}

p1 = p1->next;

p2 = p2->next;

}

secondHalf = reverseList(secondHalf);

if (mid != NULL) {

prev\_slow->next = mid;

mid->next = secondHalf;

} else {

prev\_slow->next = secondHalf;

}

return isPalin;

}

void DisplayList(Node\* head) {

Node\* current = head;

while (current != NULL) {

cout << current->data << " -> ";

current = current->next;

}

cout << "NULL!" <<endl;

}

int main() {

int arr[] = {1, 2, 3, 4, 3, 2, 1};

int n = sizeof(arr) / sizeof(arr[0]);

Node\* head = SLL(arr, n);

cout << "Linked List: ";

DisplayList(head);

if (isPalindrome(head)) {

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

} else {

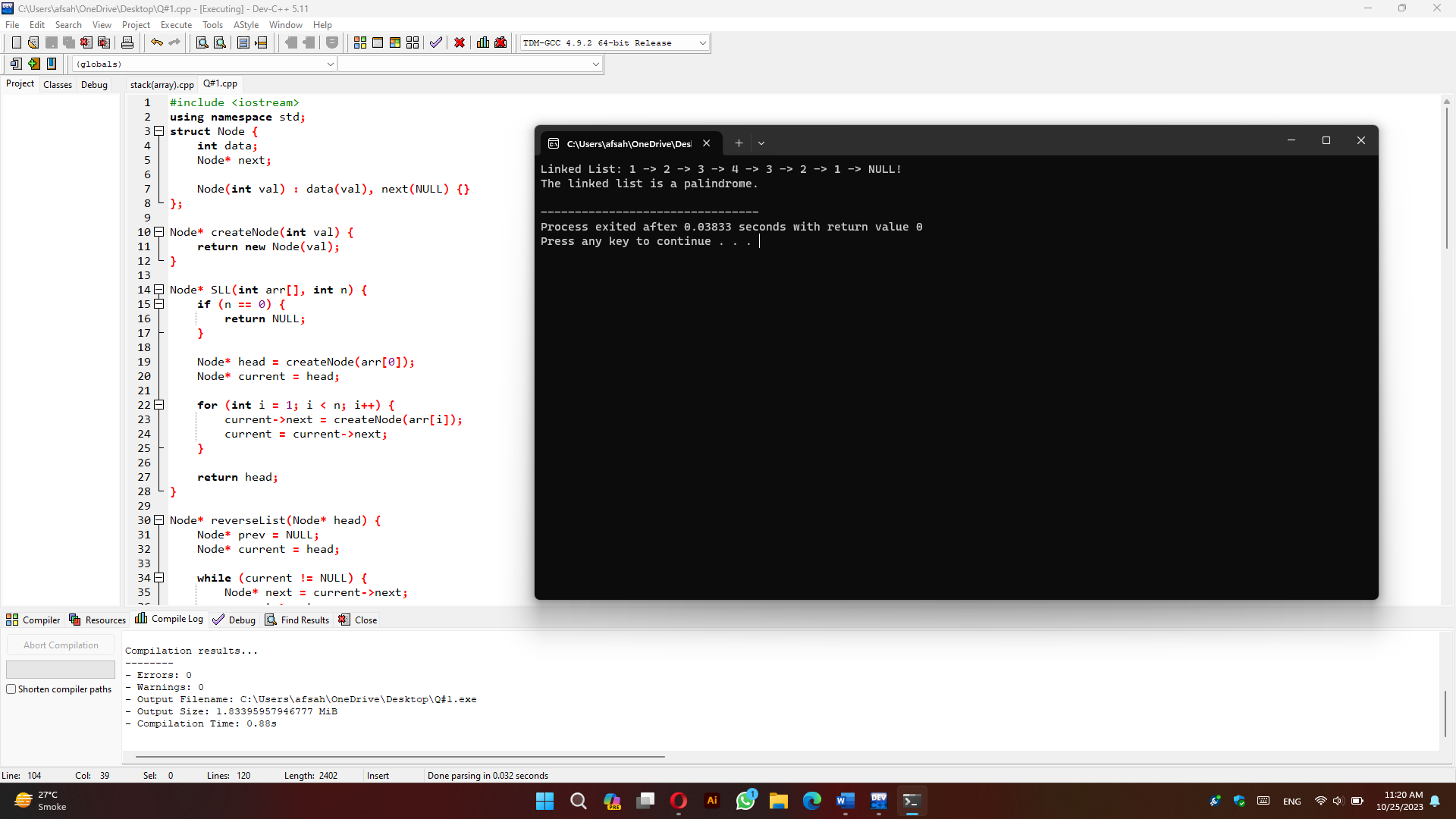
cout << "The linked list is not a palindrome." << endl;

}

return 0;

}

**OUTPUT:**

****

**Q#2:**

**Stack implementation using Array**

**Code:**

#include <iostream>

using namespace std;

int stack[100];

int n=100;

int top=-1;

void push(int val){

if(top>n-1){

cout<<"Stack overflow!"<<endl;

}

else{

top=top+1;

stack[top]=val;

}

}

void pop(){

if(top==-1){

cout<<"Stack underflow!"<<endl;

}

else{

cout<<"Top of stack:"<<stack[top]<<endl;

top=top-1;

stack[top];

}

}

void display(){

if (top<0) {

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

} else {

cout <<"Stack elements after pop: ";

for(int i=0; i<=top; i++){

cout<<stack[i]<<" ";

}

cout<<endl;

}

}

int main(){

push(1);

push(2);

push(3);

push(4);

push(5);

display();

pop();

display();

pop();

display();

pop();

display();

pop();

display();

pop();

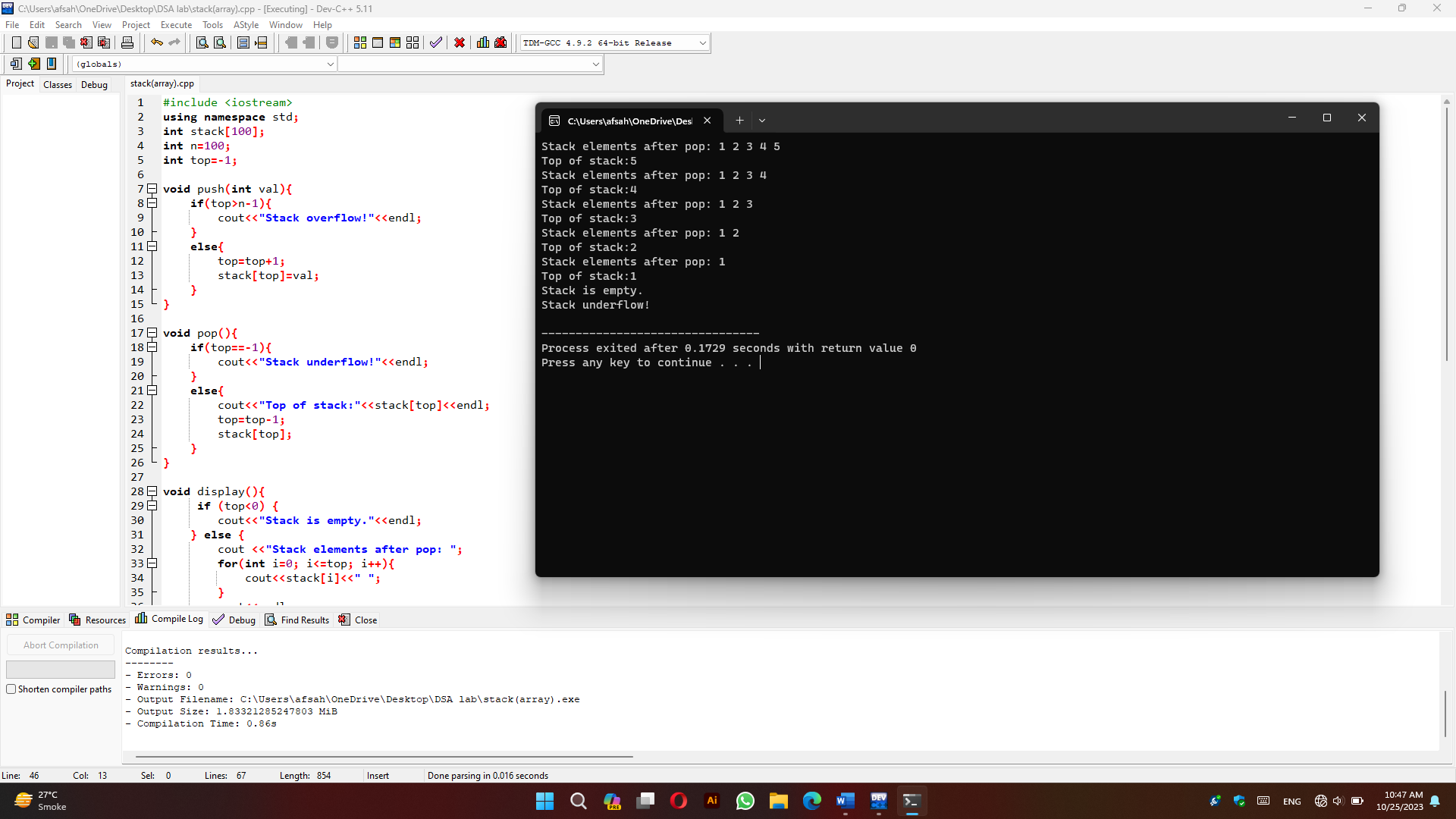
display();

pop();

return 0;

}

**OUTPUT:**

****