|  |  |
| --- | --- |
| download | COMSATS University Islamabad, Vehari Campus Department of Computer Science |

**Class: BCS-SP22-4A Title : Lab Mid Programs**

**Subject: Data Structures and Algorithms-Lab Instructor: Yasmeen Jana**

**Name: Ahmad Hussain Reg. No: SP22-BCS-008**

**QUESTION # 01**

**Code:**

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

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

};

class SinglyLinkedList {

public:

Node\* head;

SinglyLinkedList() : head(NULL) {}

void insertEnd(int data) {

Node\* newNode = new Node(data);

if (head == NULL) {

head = newNode;

} else {

Node\* current = head;

while (current->next != NULL) {

current = current->next;

}

current->next = newNode;

}

}

void reverse() {

Node\* prev = NULL;

Node\* current = head;

Node\* nextNode = NULL;

while (current != NULL) {

nextNode = current->next;

current->next = prev;

prev = current;

current = nextNode;

}

head = prev;

}

bool isPalindrome() {

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

return true;

}

Node\* slow = head;

Node\* fast = head;

Node\* prevSlow = NULL;

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

fast = fast->next->next;

prevSlow = slow;

slow = slow->next;

}

prevSlow->next = NULL;

reverse();

Node\* firstHalf = head;

Node\* secondHalf = slow;

while (firstHalf != NULL && secondHalf != NULL) {

if (firstHalf->data != secondHalf->data) {

return false;

}

firstHalf = firstHalf->next;

secondHalf = secondHalf->next;

}

return true;

}

};

int main() {

SinglyLinkedList list;

list.insertEnd(1);

list.insertEnd(2);

list.insertEnd(3);

list.insertEnd(3);

list.insertEnd(2);

list.insertEnd(1);

if (list.isPalindrome()) {

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

} else {

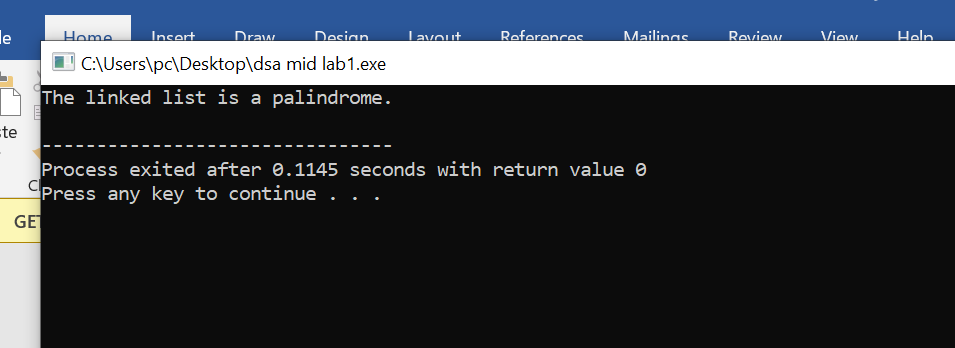
cout << "The linked list is not a palindrome." << endl;

}

return 0;

}

**Screenshorts:**

****

**QUESTION # 02**

Code**:**

#include <iostream>

using namespace std;

int stack[50], s=50, top=-1;

void push(int value) {

if(top>s-1)

cout<<"Stack is full "<<endl;

else {

top++;

stack[top]=value;

}

}

void pop() {

if(top<=-1)

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

else {

cout<<"The element that is poped from stack "<< stack[top] <<endl;

top--;

}

}

void display() {

if(top>=0) {

cout<<"Elements in stack are : ";

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

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

cout<<endl;

} else

cout<<"Stack is empty";

}

int main(){

int choice, value;

cout<<"1) Push in stack"<<endl;

cout<<"2) Pop from stack"<<endl;

cout<<"3) Display stack"<<endl;

cout<<"4) To Exit the program"<<endl;

do {

cout<<"Enter your choice to perform above operations: "<<endl;

cin>>choice ;

switch(choice ) {

case 1: {

cout<<"Enter value to be pushed in stack :"<<endl;

cin>>value;

push(value);

break;

}

case 2: {

pop();

break;

}

case 3: {

display();

break;

}

case 4: {

cout<<"program is terminated successfully"<<endl;

break;

}

default: {

cout<<"you have entered an Invalid Choice"<<endl;

}

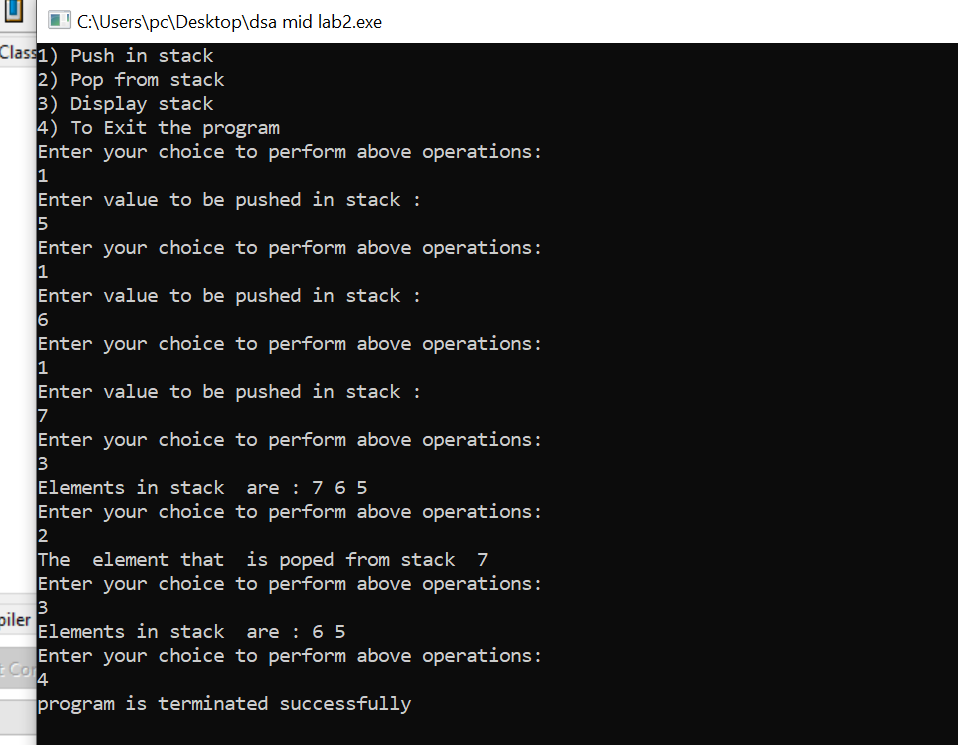
}

}while(choice!=4);

return 0;

}

**Screenshorts:**

****