**Name: Zubair Farooq**

**Roll no: SP22-BCS-004(A)**

**Subject: Data Structures Lab**

**Teacher: Mam Yasmeen**

**CUI Vehari**

**Question 1:Stack using array.**

**Code:**

**#include <iostream>**

**using namespace std;**

**int stack[80], n=80, top=-1;**

**void push(int val) {**

**if(top>n-1)**

**cout<<"Stack Overflow"<<endl;**

**else {**

**top++;**

**stack[top]=val;**

**}**

**}**

**void pop() {**

**if(top<=-1)**

**cout<<"Stack Underflow"<<endl;**

**else {**

**cout<<"The popped element is "<< stack[top] <<endl;**

**top--;**

**}**

**}**

**void display() {**

**if(top>=0) {**

**cout<<"Stack elements are:";**

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

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

**cout<<endl;**

**} else**

**cout<<"Stack is empty";**

**}**

**int main() {**

**int ch, val;**

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

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

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

**cout<<"4) Exit"<<endl;**

**do {**

**cout<<"Enter choice: "<<endl;**

**cin>>ch;**

**switch(ch) {**

**case 1: {**

**cout<<"Enter value to be pushed:"<<endl;**

**cin>>val;**

**push(val);**

**break;**

**}**

**case 2: {**

**pop();**

**break;**

**}**

**case 3: {**

**display();**

**break;**

**}**

**case 4: {**

**cout<<"Exit"<<endl;**

**break;**

**}**

**default: {**

**cout<<"Invalid Choice"<<endl;**

**}**

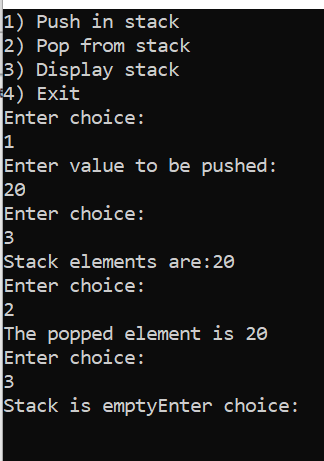
**}**

**}while(ch!=4);**

**return 0;**

**}**

**Output:**



**Question 2:Check Palindrome or not.**

**Code:**

**#include <iostream>**

**using namespace std;**

**struct Node {**

**int data;**

**Node\* next;**

**};**

**Node\* createNode(int data) {**

**Node\* newNode = new Node;**

**newNode->data = data;**

**newNode->next = NULL;**

**return newNode;**

**}**

**void insertAtEnd(Node\*& head, int data) {**

**Node\* newNode = createNode(data);**

**if (head == NULL) {**

**head = newNode;**

**} else {**

**Node\* current = head;**

**while (current->next != NULL) {**

**current = current->next;**

**}**

**current->next = newNode;**

**}**

**}**

**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;**

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

**slow = slow->next;**

**fast = fast->next->next;**

**}**

**Node\* secondHalf = reverseList(slow->next);**

**Node\* firstHalf = head;**

**while (secondHalf != NULL) {**

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

**return false;**

**firstHalf = firstHalf->next;**

**secondHalf = secondHalf->next;**

**}**

**return true;**

**}**

**void displayList(Node\* head) {**

**Node\* current = head;**

**while (current != NULL) {**

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

**current = current->next;**

**}**

**cout << "NULL" << endl;**

**}**

**int main() {**

**Node\* head = NULL;**

**insertAtEnd(head, 1);**

**insertAtEnd(head, 2);**

**insertAtEnd(head, 2);**

**insertAtEnd(head, 1);**

**cout << "Linked List: ";**

**displayList(head);**

**if (isPalindrome(head)) {**

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

**} else {**

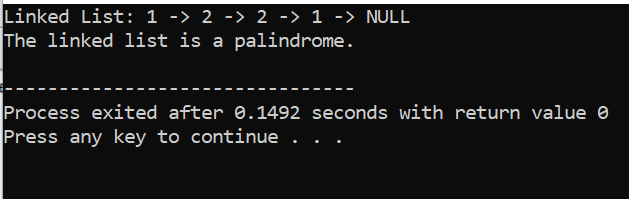
**cout << "The linked list is not a palindrome." << endl;**

**}**

**return 0;**

**}**

**Output:**



Thank you!