|  |  |
| --- | --- |
| **MID TERM** | |
| **Name** | **Hafiz Shahid Iqbal** |
| **Registration No** | **SP22-BCS-017** |
| **Section** | **A** |
| **Subject** | **Data Structure lab** |
| **Submitted to** | **Mam Yasmeen Jana** |
| **Date** | **25/10/2023** |
| **Comsats University Vehari** | |

**Question no 1:**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

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

};

void insert(Node\*& head, 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 printList(Node\* head) {

Node\* current = head;

while (current != NULL) {

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

current = current->next;

}

cout << "NULL" << std::endl;

}

Node\* reverseList(Node\* head) {

Node\* prev = NULL;

Node\* current = head;

Node\* next;

while (current != NULL) {

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;

}

int main() {

Node\* head = NULL;

insert(head, 1);

insert(head, 2);

insert(head, 3);

insert(head, 2);

insert(head, 1);

cout << "Linked List: ";

printList(head);

if (isPalindrome(head)) {

std::cout << "The linked list is a palindrome.\n";

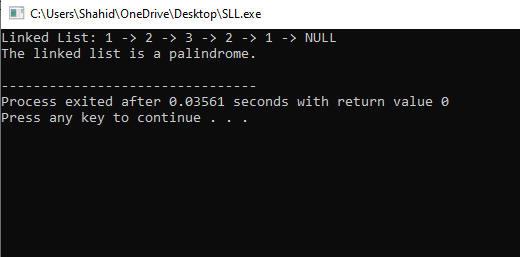
} else {

std::cout << "The linked list is not a palindrome.\n";

}

return 0;

}



**Question no 2:**

#include <iostream>

using namespace std;

int stack[100], n=100, 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;

}

