**Task-1**

#include <iostream>

using namespace std;

class stack

{

private:

int top;

int \*arr;

int s;

public:

stack(int size = 10) {

s = size;

arr = new int[size];

top = -1;

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

arr[i] = 0;

}

}

bool isfull() {

if (top == s-1) {

return true;

}

else

return false;

}

bool isEmpty(){

if (top == -1) {

return true;

}

else {

return false;

}

}

void push(int val)

{

if (isfull()){

cout << "Stack Overflow\n";

}

else {

top++;

arr[top] = val;

}

}

int pop()

{

if (isEmpty())

{

cout << "Stack Underflow\n";

}

else

{

int popValue = arr[top];

arr[top] = 0;

top--;

return popValue;

}

}

int topElement() {

return arr[top];

}

void display() {

int c = 0;

while (c<top){

cout << arr[c] << endl;

c++;

}

}

};

int main(){

int val, choice;

stack st;

do {

std::cout << "1-Push\n2-Pop\n3-IsEmpty\n4-IsFUll\n5-Get Top\n6-Display\nChoose Option:";

std::cin >> choice;

switch (choice) {

case 1:

std::cout << "Enter a Value: ";

std::cin >> val;

st.push(val);

break;

case 2:

std::cout << "Value: " << st.pop() << "\n";

break;

case 3:

std::cout << "Isempty: " << st.isEmpty() << "\n";

break;

case 4:

std::cout << "IsFUll: " << st.isfull() << "\n";

break;

case 5:

std::cout << "Top Element: " << st.topElement() << "\n";

break;

case 6:

st.display();

break;

}

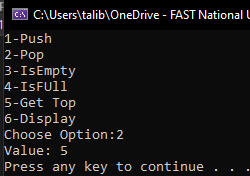
system("pause");

system("cls");

} while (choice != 0);

}

Text

Description automatically generated ****

**Text

Description automatically generated**

**Task-2**

#include<iostream>

struct Node {

int data;

Node\* next;

};

class stack {

Node \*top;

public:

stack() {

top = NULL;

}

bool Isempty() {

if (top == NULL)

return true;

else

return false;

}

int pop() {

if (Isempty()) {

std::cout << "Stack underflow\n";

return 0;

}

else {

int popvalue = top->data;

Node\* curr = top;

top = top->next;

delete curr;

curr = NULL;

return popvalue;

}

}

void push(int val) {

Node\* newNode = new Node;

newNode->data = val;

if (top == NULL) {

top = newNode;

top->next = NULL;

}

else

newNode->next = top;

top = newNode;

}

};

int main() {

int val,choice;

stack st;

do{

std::cout << "1-Push\n2-Pop\n3-IsEmpty\nChoose Option:";

std::cin >> choice;

switch (choice){

case 1:

std::cout << "Enter a Value: ";

std::cin >> val;

st.push(val);

break;

case 2:

std::cout << "Value: " << st.pop() << "\n";

break;

case 3:

std::cout << "Isempty: " << st.Isempty() << "\n";

break;

}

system("pause");

system("cls");

} while (choice!=0);

}

Text

Description automatically generated Text

Description automatically generatedText

Description automatically generated

**Task-3**

#include <iostream>

#include<string>

using namespace std;

class stack

{

private:

int top;

int\* arr;

int s;

public:

stack(int size = 10) {

s = size;

arr = new int[size];

top = -1;

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

arr[i] = 0;

}

}

bool isfull() {

if (top == s - 1) {

return true;

}

else

return false;

}

bool isEmpty() {

if (top == -1) {

return true;

}

else {

return false;

}

}

void push(int val)

{

if (isfull()) {

cout << "Stack Overflow\n";

}

else {

top++;

arr[top] = val;

}

}

int pop()

{

if (isEmpty())

{

cout << "Stack Underflow\n";

}

else

{

int popValue = arr[top];

arr[top] = 0;

top--;

return popValue;

}

}

int topElement() {

return arr[top];

}

void display() {

int c = 0;

while (c < top) {

cout << arr[c] << endl;

c++;

}

}

};

bool ispalindrome(stack\* stack1, stack\* stack2) {

bool check = false;

while (stack1->isEmpty()!=1){

if (stack1->pop() == stack2->pop())

check = true;

else

{

check = false;

break;

}

}

return check;

}

int main() {

string str;

cout << "ENter String: ";

getline(cin, str);

stack stack1,stack2;

int size = str.size()-1;

int c = 0,end = size;

while (c<size+1){

stack1.push(str[c]);

stack2.push(str[end]);

c++;

end--;

}

cout<<"Ispalindrome: "<< ispalindrome(&stack1, &stack2);

}

