|  |  |  |
| --- | --- | --- |
| Image result for latest marwadi university logo | **Marwadi University**  **Faculty of Technology**  **Department of Information and Communication Technology** | |
| **Subject: DSC  (01CT0308)** | Aim: Implementations of stack using the menu-driven program. | |
| **Experiment No: 2** | **Date: 01- 09 - 2023** | **Enrolment No:-** 92200133030 |

**Experiment – 2**

**Objective:** Implementations of stack using the menu-driven program.

**Code :-**

1. **Using Array**

#include <iostream>

using namespace std;

class Stack {

private:

static const int MAX\_SIZE = 100;

int arr[MAX\_SIZE];

int top;

public:

Stack() {

top = -1;}

bool isEmpty() {

return top == -1; }

bool isFull() {

return top == MAX\_SIZE - 1;}

void push(int value) {

if (isFull()) {

cout << "Stack is full. Cannot push more elements." << endl;

} else {

arr[++top] = value;

cout << "Pushed " << value << " onto the stack." << endl;

}}

void pop() {

if (isEmpty()) {

cout << "Stack is empty. Cannot pop elements." << endl;

} else {

cout << "Popped " << arr[top--] << " from the stack." << endl;

}}

void display() {

if (isEmpty()) {

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

} else {

cout << "Stack elements: ";

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

cout << arr[i] << " ";}

cout << endl;

}}};

int main() {

Stack stack;

int choice, value ,size;

cout<< "Choose The Number To Perform Various Operartion On Your Stack :- " << endl;

do {

cout << "Stack Menu:" << endl;

cout << "1. Push" << endl;

cout << "2. Pop" << endl;

cout << "3. Check Underflow" <<endl ;

cout << "4. Check Overflow" << endl ;

cout << "5. Display" << endl;

cout << "6. Quit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter value to push: ";

cin >> value;

stack.push(value);

break;

case 2:

stack.pop();

break;

case 3:

stack.isEmpty();

break;

case 4:

stack.isFull();

break;

case 5:

stack.display();

break;

case 6:

cout << "Exiting program." << endl;

break;

default:

cout << "Invalid choice. Please select a valid option." << endl;

break;

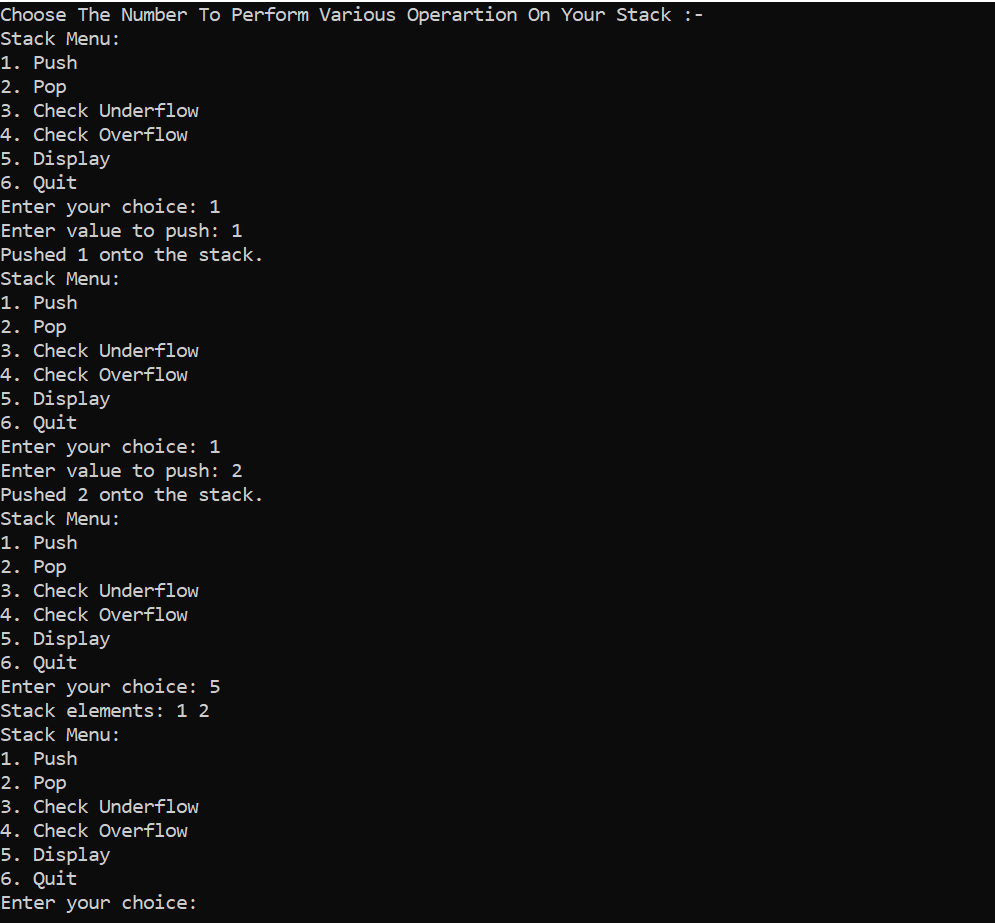
}

} while (choice != 4);

return 0;

}

**Output:**



1. **Using LinkedList**

#include <iostream>

using namespace std;

class node {

public:

int data;

node\* next;

node(int val) {

data = val;

next = NULL;

}};

class Stack {

public:

node\* top;

Stack() {

top = NULL;}

void push(int val) {

cout << val << " Is Pushed In Your Stack." << endl;

node\* n = new node(val);

if (top == NULL) {

node\* n = new node(val);

n->next = top;

top = n;

return;}

node\* temp = top;

while (temp->next != NULL) {

temp = temp->next;}

temp->next = n;}

int gettop() {

if (top == NULL) {

return -1;}

node\* temp = top;

int topindex = 0;

while (temp != NULL) {

topindex++;

temp = temp->next;}

return topindex;}

void pop() {

if (top == NULL) {

cout << "Stack Is Empty." << endl;

return;}

if (top->next == NULL) {

cout << top->data << " Is Popped From Your Stack." << endl;

delete top;

top = NULL;}

else {

node\* temp = top;

while (temp->next->next != NULL) {

temp = temp->next;}

node\* todelete = temp->next;

temp->next = nullptr;

cout << todelete->data << " Is Popped From Your Stack." << endl;

delete todelete;}}

bool isEmpty() {

return gettop() == -1;}

bool isfull(int size) {

return gettop() >= size – 1;}

void display() {

node\* temp = top;

while (temp != NULL) {

cout << temp->data << " -> ";

temp = temp->next;}

cout << " NULL " << endl;}};

int main() {

Stack stack;

int choice, value, size;

cout << "Enter The Size Of Your Stack." << endl;

cin >> size;

do {

cout << "Stack Menu:" << endl;

cout << "1. Push" << endl;

cout << "2. Pop" << endl;

cout << "3. Check Underflow" << endl;

cout << "4. Check Overflow" << endl;

cout << "5. Display" << endl;

cout << "6. Quit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

if (stack.gettop() >= size) {

cout << "Stack Is Full." << endl;

} else {

cout << "Enter value to push: ";

cin >> value;

stack.push(value);

}

break;

case 2:

stack.pop();

break;

case 3:

if (stack.isEmpty()) {

cout << "Stack Is Empty." << endl;

} else {

cout << "Stack Is Not Empty.";

}

break;

case 4:

if (stack.isfull(size)) {

cout << "Stack Is Full." << endl;

} else {

cout << "Stack Is Not Full.";

}

break;

case 5:

stack.display();

break;

case 6:

cout << "Exiting program." << endl;

break;

default:

cout << "Invalid choice. Please select a valid option." << endl;

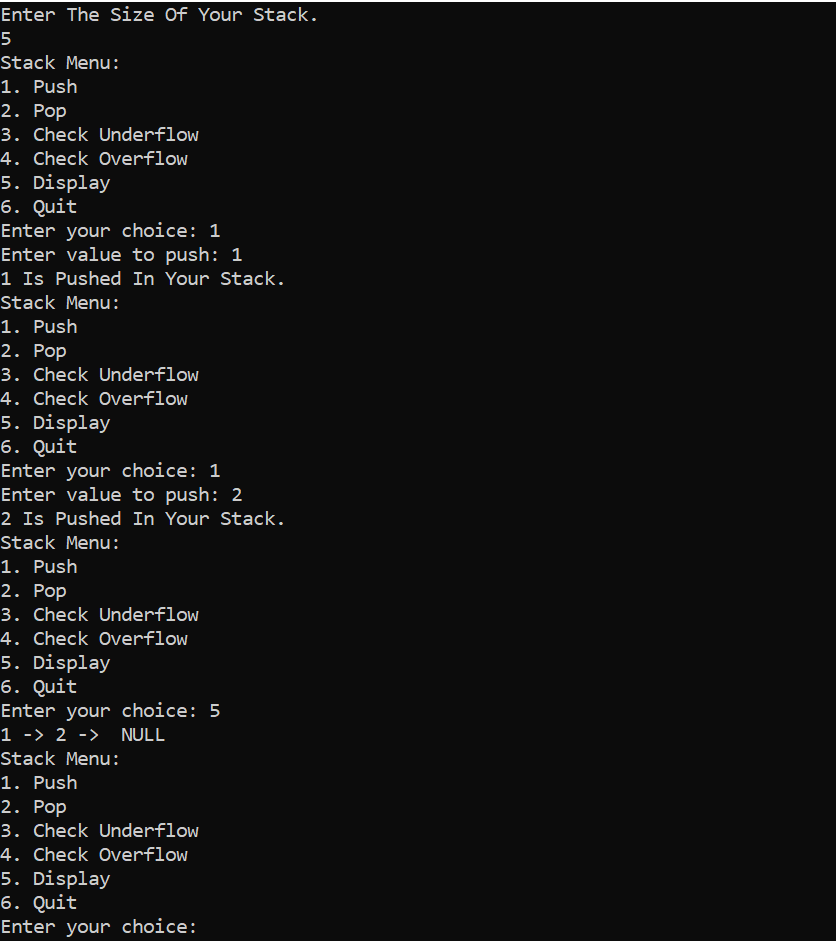
}

} while (choice != 6);

return 0;

}

**Output:**

****