

// Write a program in C++ to Implement stack as an ADT using singly linked list and use this ADT to perform different operations on it.
(Push, Pop, IsEmpty, Peek, IsFull, Display)
// Using C++

```
#include <iostream>
#include <stdlib.h>
using namespace std;

struct Node
{
    int data;
    struct Node *next;
};

struct Node *top = NULL;

void push(int val)
{
    struct Node *newnode = (struct Node *)malloc(sizeof(struct Node));
    if (newnode == NULL)
    {
        cout << "\n Stack Overflow (No memory available)";
        return;
    }
    newnode->data = val;
    newnode->next = top;
    top = newnode;
    cout << "\n " << val << " pushed into the stack.";
}

void pop()
{
    if (top == NULL)
        cout << "\n Stack Underflow (No elements to pop)";
    else
    {
        cout << "\n The popped element is: " << top->data;
        top = top->next;
    }
}

void peek()
{
    if (top == NULL)
        cout << "\n Stack is empty (No top element)";
    else
```

```

        cout << "\n Top element is: " << top->data;
    }

bool isEmpty()
{
    return (top == NULL);
}

bool isFull()
{
    struct Node *temp = (struct Node *)malloc(sizeof(struct Node));
    if (temp == NULL)
        return true;
    free(temp);
    return false;
}

void display()
{
    struct Node *ptr;
    if (top == NULL)
        cout << "\n Stack is empty.";
    else
    {
        ptr = top;
        cout << "\n Stack elements are: ";
        while (ptr != NULL)
        {
            cout << ptr->data << " ";
            ptr = ptr->next;
        }
        cout << endl;
    }
}

int main()
{
    int ch, val;
    cout << "\n 1) Push in stack";
    cout << "\n 2) Pop from stack";
    cout << "\n 3) Peek top element";
    cout << "\n 4) Check if stack is empty";
    cout << "\n 5) Check if stack is full";
    cout << "\n 6) Display stack";
    cout << "\n 7) Exit";

    do
    {

```

```

cout << "\n\n Enter choice: ";
cin >> ch;
switch (ch)
{
case 1:
    cout << "\n Enter value to be pushed: ";
    cin >> val;
    push(val);
    break;

case 2:
    pop();
    break;

case 3:
    peek();
    break;

case 4:
    if (isEmpty())
        cout << "\n Stack is Empty.";
    else
        cout << "\n Stack is NOT Empty.";
    break;

case 5:
    if (isFull())
        cout << "\n Stack is Full (Memory allocation failed).";
    else
        cout << "\n Stack is NOT Full.";
    break;

case 6:
    display();
    break;

case 7:
    cout << "\n Exiting program...";
    break;

default:
    cout << "\n Invalid choice!";
}
} while (ch != 7);
return 0;
}

```