PRN: 2020BTEIT00041

Stack ADT:

```cpp
/*
    Name: Om Vivek Gharge
    PRN: 2020BTEIT00041
*/
// Stack ADT :

#include <bits/stdc++.h>

using namespace std;

class Node{
public:
    int data;
    Node* next;

    Node(){
        this->data = 0;
        this->next = NULL;
    }

    Node(int data){
        this->data = data;
        this->next = NULL;
    }
};

// template

template <class stack>
class Stack{
private:
    Node* top = NULL;
    int count = 0;
public:
    // Constructor
    Stack(){
        this->top = NULL;
        this->count = 0;
    };

    // Destructor
    ~Stack(){
        Node* temp = this->top;
        while(temp != NULL){
            Node* temp2 = temp->next;
            delete temp;
            temp = temp2;
        }
```

```
49      };
50
51      void Push(int data);
52      int Pop();
53      void Display();
54  };
55
56  // Push
57  template<>
58  void Stack<Node>::Push(int data){
59
60      // Create a new node and store the data
61      Node* newNode = new Node(data);
62
63      // If stack is empty, then newNode will be the head
64      if(this->top == NULL){
65          top = newNode;
66      }
67
68      // If stack is not empty, then add the newNode to the top
69      else{
70          newNode->next = this->top;
71          this->top = newNode;
72      }
73
74      // Increment the count
75      count++;
76  }
77
78  // Pop
79  template<>
80  int Stack<Node>::Pop(){
81      // If stack is empty, then return -1
82      if(top == NULL){
83          cout<<"Stack Underflow\n";
84          return -1;
85      }
86
87      // If stack is not empty, then return the top and delete the top
88      else{
89          // Store the top data
90          int data = top->data;
91
92          // Store the top's next node
93          Node* temp = top;
94
95          // move the top to the next node
96          top = top->next;
```

```
        // Delete the top
        delete temp;

        // Decrement the count
        count--;

        // Return the top data
        return data;
    }
}

// Display
template<>
void Stack<Node>::Display(){
    // If stack is empty, then print "Stack is empty"
    if(top == NULL){
        cout<<"Stack is empty\n";
    }

    // If stack is not empty, then print the stack
    else{
        // Create a new node and store the top
        Node* temp = top;
        cout<<"\nStack:\n";
        // Print the stack
        while(temp != NULL){
            cout<<"\n"<<temp->data;
            temp = temp->next;
        }
        cout<<endl;
    }
}

int main(){
    Stack<Node> s;

    // Menu for Stack ADT
    int choice;
    while(1){
        cout << "-------------MENU-------------\n";
        cout << "1. Push" << endl;
        cout << "2. Pop" << endl;
        cout << "3. Display" << endl;
        cout << "4. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
```

```
143         cin >> choice;
144
145         switch(choice){
146             case 1:
147                 int data;
148                 cout << "Enter data: ";
149                 cin >> data;
150                 s.Push(data);
151                 break;
152             case 2:
153                 s.Pop();
154                 break;
155             case 3:
156                 s.Display();
157                 break;
158             case 4:
159                 cout << "Exiting..." << endl;
160                 return 0;
161             default:
162                 cout << "Invalid choice" << endl;
163         }
164     }
165     return 0;
166 }
```

OUTPUT:

```
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter data:
1
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter data: 2
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter data: 3
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3

Stack:

3
2
1
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
------------MENU------------
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3

Stack:

2
1
```