PRN: 2020BTEIT00041

Stack Implementation using Array:

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;

class Stack{

public:
    int size;
    int top;
    int* arrStack;

    Stack(int size){
        this->size = size;
        top = -1;
        arrStack = new int[size];
    }
};

// Push
void Push(Stack* stack, int data){
    // Check if stack is full
    if(stack->top == stack->size - 1) cout<<"Stack Overflow \n";
    else{
        stack->arrStack[++stack->top] = data;
    }
}

// Pop
void Pop(Stack* stack){
    // Check if stack is empty
    if(stack->top == -1) cout<<"Stack Underflow \n";
    else{
        stack->top--;
        cout<<"Popped element is "<<stack->arrStack[stack->top+1]<<"\n"
    }
}

// Peek
void Peek(Stack* stack, int index){
    // Check if indez is valid or not
    if(index > stack->top + 1 || index < 0) cout<<"Invalid index \n";
    else{
        cout<<"Element at index "<<index<<" is "<<stack->arrStack[stack
    }
}

// isEmpty
bool isEmpty(Stack* stack){
    return stack->top == -1;
```

```
50  }
51
52  // isFull
53  bool isFull(Stack* stack){
54      return stack->top == stack->size - 1;
55  }
56
57  // Stack Top
58  int StackTop(Stack* stack){
59      if(!isEmpty(stack)) return stack->arrStack[stack->top];
60      else return -1;
61  }
62
63  //Print the stack
64  void Display(Stack* stack){
65      if(stack->top == -1) cout<<"Stack is empty. \n";
66      else{
67          cout<<"Stack is: ";
68          for(int i = stack->top; i >= 0; i--){
69              cout<<stack->arrStack[i]<<" ";
70          }
71          cout<<"\n";
72      }
73  }
74
75  int main(){
76      int n;
77      cout<<"Enter the size of the stack: ";
78      cin>>n;
79
80      // Create a stack of size n
81      Stack* stack = new Stack(n);
82
83      // Menu for stack operations
84      int choice;
85      do{
86          cout<<"\n1. Push \n2. Pop \n3. Peek \n4. Display \n5. Exit \nE
87          cin>>choice;
88
89          switch(choice){
90              case 1:
91                  int data;
92                  cout<<"Enter the data to be pushed: ";
93                  cin>>data;
94                  Push(stack, data);
95                  break;
```

```
            case 2:
                Pop(stack);
                break;
            case 3:
                int index;
                cout<<"Enter the index of the element to be peeked: ";
                cin>>index;
                Peek(stack, index);
                break;
            case 4:
                Display(stack);
                break;
            case 5:
                cout<<"Exiting... \n";
                break;
            default:
                cout<<"Invalid choice. \n";
        }
    }while(choice != 5);

    return 0;
}
```

OUTPUT:

```
Enter the size of the stack: 5

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter the data to be pushed: 1

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter the data to be pushed: 2

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter the data to be pushed: 3

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter the data to be pushed: 4

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter the data to be pushed: 5

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 4
Stack is: 5 4 3 2 1

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 2
Popped element is 5

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 4
Stack is: 4 3 2 1

1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 5
Exiting...
```