FAIZAN CHOUDHARY

20BCS021

DSA LAB

2nd November 2021

CODE: (code pasted in this format for readability)

```
#include <iostream>
using namespace std;
const int LIMIT=20;
int top=-1;

int *stack = (int *) malloc (LIMIT * sizeof(int));

int isEmpty ()
{
    if (top==-1)
     return 1;
    else
     return 0;
}

int isFull ()
{
    if (top==(LIMIT-1))
     return 1;
    else
     return 0;
}

void display ()
{
    if (isEmpty()==1)
     cout<<"\nStack is empty! Nothing to display\n";
    else
    {
        cout<<endl<<stack[top]<<"  <--"<<endl;
        for (int i=top-1; i>=0; i--)
         cout<<stack[i]<<endl;
    }
}

int size ()
{
    if (isEmpty()==1)
     return 0;
    else
```

```
     return (top+1);
}

void peek ()
{
    if (isEmpty()==1)
     cout<<"\nStack is empty..."<<endl;
    else
     cout<<"\nTop element is: "<<stack[top]<<endl;
}

void push (int n)
{
    if (isFull()==1)
     cout<<"\nStack Overflow! Maximum limit reached..."<<endl;
    else
    {
        top++;
        stack[top]=n;
        display();
    }
}

void pop ()
{
    if (isEmpty()==1)
     cout<<"\nStack Underflow! Stack is empty..."<<endl;
    else
    {
        cout<<"\nPopping top element: "<<stack[top]<<endl;
        top--;
        display();
    }
}

int main()
{
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";

    int ch,n,num, *stack1;

    cout<<"\nEnter number of elements initially: ";
    cin>>num;
    stack1 = (int *) malloc (num * sizeof(int));
    if (stack1==NULL)
    {
        cout<<"\nMemory could not be allocated!";
        exit(1);
    }
    stack = stack1;

    while (true)
    {
        A:
```

```
        cout<<"\nMENU:\n1. Push into stack\n2. Pop element\n3. Peek top element\n4. Check
if stack is full\n5. Check if stack is empty\n6. Size of the stack\n7. Display stack\n8.
Exit\n";
        cin>>ch;
        switch (ch)
        {
        case 1: cout<<"Enter the element to be pushed: ";
                cin>>n;
                push(n);
                break;
        case 2: pop();
                break;
        case 3: peek();
                break;
        case 4: if (isFull()==1)
                 cout<<"\nStack is full!\n";
                else
                 cout<<"\nStack is not full.\n";
                break;
        case 5: if (isEmpty()==1)
                 cout<<"\nStack is empty!\n";
                else
                 cout<<"\nStack is not empty.\n";
                break;
        case 6: cout<<"\nSize of the stack is: "<<size()<<endl;
                break;
        case 7: cout<<"\nStack elements: "<<endl;
                display();
                break;
        case 8: exit(0);
        default: cout<<"\nWrong choice! Enter again...\n";
                 goto A;
        }
    }
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

Enter number of elements initially: 5

MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
1
Enter the element to be pushed: 55

55  <--
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
1
Enter the element to be pushed: 44

44  <--
55
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
1
Enter the element to be pushed: 33

33  <--
44
55
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
3

Top element is: 33
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
4

Stack is not full.
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
6

Size of the stack is: 3
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
2

Popping top element: 33

44  <--
55
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
2

Popping top element: 44

55  <--
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
2

Popping top element: 55

Stack is empty! Nothing to display
```

```
MENU:
1. Push into stack
2. Pop element
3. Peek top element
4. Check if stack is full
5. Check if stack is empty
6. Size of the stack
7. Display stack
8. Exit
5

Stack is empty!
```