

Ex. No. 2

19-07-2017

ARRAY IMPLEMENTATION OF STACK**Question**

Demonstrate the working of a stack of size N using an array. The elements of the stack may be assumed to be of type integer. The three operations to be supported are: (a) Push, (b) Pop, (c) Display. The program should print appropriate messages for stack overflow, and stack underflow.

Algorithm

- 1) Start.
- 2) Create an array, size to be given by the user.
- 3) Create a switch case and display the menu, get the operation to be done by the user and perform accordingly.
- 4) In case 1, it will perform the push operation to insert an element as:
Check for $\text{top} = \text{size} - 1$, if its -1, print stack is full.
Else increment top and insert the data in the stack as $\text{stk}[\text{top}] = \text{item}$.
- 5) In case 2, it will perform the pop operation to delete the element as:
If $\text{top} = -1$, print stack is empty.
Else $\text{stk}[\text{top}]$ is popped, and top is decremented.
- 6) In case 3, check for $\text{top} = -1$, print stack is empty, else display the stack.
- 7) End.

Program

```
#include <iostream>

#include<stdlib.h>

using namespace std;
```

```
int main()
{
    int a;

    cout<<"\n Enter the size of the array: ";

    cin>>a;

    int stk[a],n,d,i;

    int top=-1;

    while(1)
    {
        cout<<"\n Choices of Stack operation:";

        cout<<"\n\n 1.Push";

        cout<<"\n 2.Pop";

        cout<<"\n 3.Display";

        cout<<"\n 4.Exit";

        cout<<"\n\n Enter the operation to be done:";

        cin>>n;

        switch(n)
        {
            case 1:

                if(top==a-1)
                {
                    cout<<"\n Stack is full"<<endl; }
                }
```

```
    else{
        cout<<"\n Enter the element:";
        cin>>d;
        top++;
        stk[top]=d;
    }

    break;
case 2:
    {
        if(top==-1)
        {
            cout<<"\n\n Stack is empty"<<endl;
        }
    }
else{
    cout<<"\n\n Popped item: "<<stk[top]<<endl;
    top--;
}
}
break;
case 3:
    if(top==-1)
    {
```

```
        cout<<"\n Stack is empty"<<endl;
    }
    else
    {
        cout<<"\n Stack elements are: ";
        for(i=top;i>=0;i--)
            cout<<stk[i]<<" ";
    }
    break;
default:
    exit(0);

}}
return 0;
}
```

Output

DATA STRUCTURES LAB

The screenshot shows the Code::Blocks IDE with a C++ project named 'main.cpp'. The code is as follows:

```
1  /*Demonstrate the working of a stack of size N using an array. The elements of the stack may be assumed to be of type integer. The
2  three operations to be supported are: (a) Push, (b) Pop, (c) Display. The program should print appropriate messages for stack
3  overflow, and stack underflow.*/
4  #include <iostream>
5  #include <stdlib.h>
6
7  using namespace std;
8
9  int main()
10 {
11     int a;
12     cout<<"\n Enter the size of the array: ";
13     cin>>a;
14     int stk[a],n,d,i;
15     int top=-1;
16     while(1)
17     {
18         cout<<"\n Choices of Stack operation:";
19         cout<<"\n\n 1.Push";
20         cout<<"\n 2.Pop";
21         cout<<"\n 3.Display";
22         cout<<"\n 4.Exit";
23         cout<<"\n\n Enter the operation to be done:";
24         cin>>n;
25         switch(n)
26         {
27             case 1:
28                 if(top==a-1)
29                 {
30
```

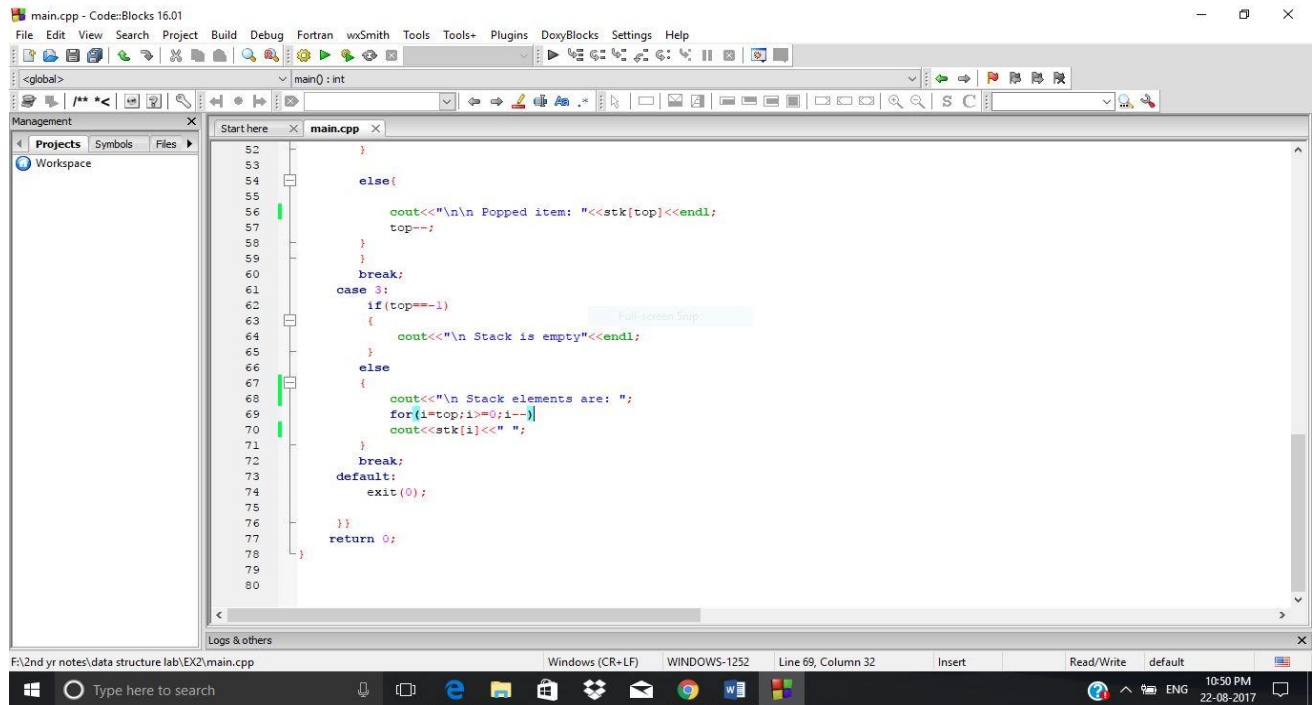
The status bar at the bottom indicates the file path is 'F:\2nd yr notes\data structure lab\EX2\main.cpp', the window title is 'WINDOWS-1252', and the cursor is at 'Line 69, Column 32'.

The screenshot shows the continuation of the C++ code in the Code::Blocks IDE:

```
29         if(top==a-1)
30         {
31             cout<<"\n Stack is full"<<endl;
32         }
33         else
34         {
35             cout<<"\n Enter the element:";
36             cin>>d;
37             top++;
38             stk[top]=d;
39             break;
40         }
41
42     case 2:
43     {
44         if(top==--1)
45         {
46             cout<<"\n\n Stack is empty"<<endl;
47         }
48         else{
49             cout<<"\n\n Popped item: "<<stk[top]<<endl;
50             top--;
51         }
52     }
53
54     case 3:
55     {
56         cout<<"\n\n Display the stack elements:";
57         for(i=0; i<=top; i++)
58             cout<<stk[i]<<" ";
59         cout<<"\n";
60     }
61
62     case 4:
63     {
64         cout<<"\n\n Exit the program.";
65         return 0;
66     }
67 }
```

The status bar at the bottom indicates the file path is 'F:\2nd yr notes\data structure lab\EX2\main.cpp', the window title is 'WINDOWS-1252', and the cursor is at 'Line 69, Column 32'.

DATA STRUCTURES LAB

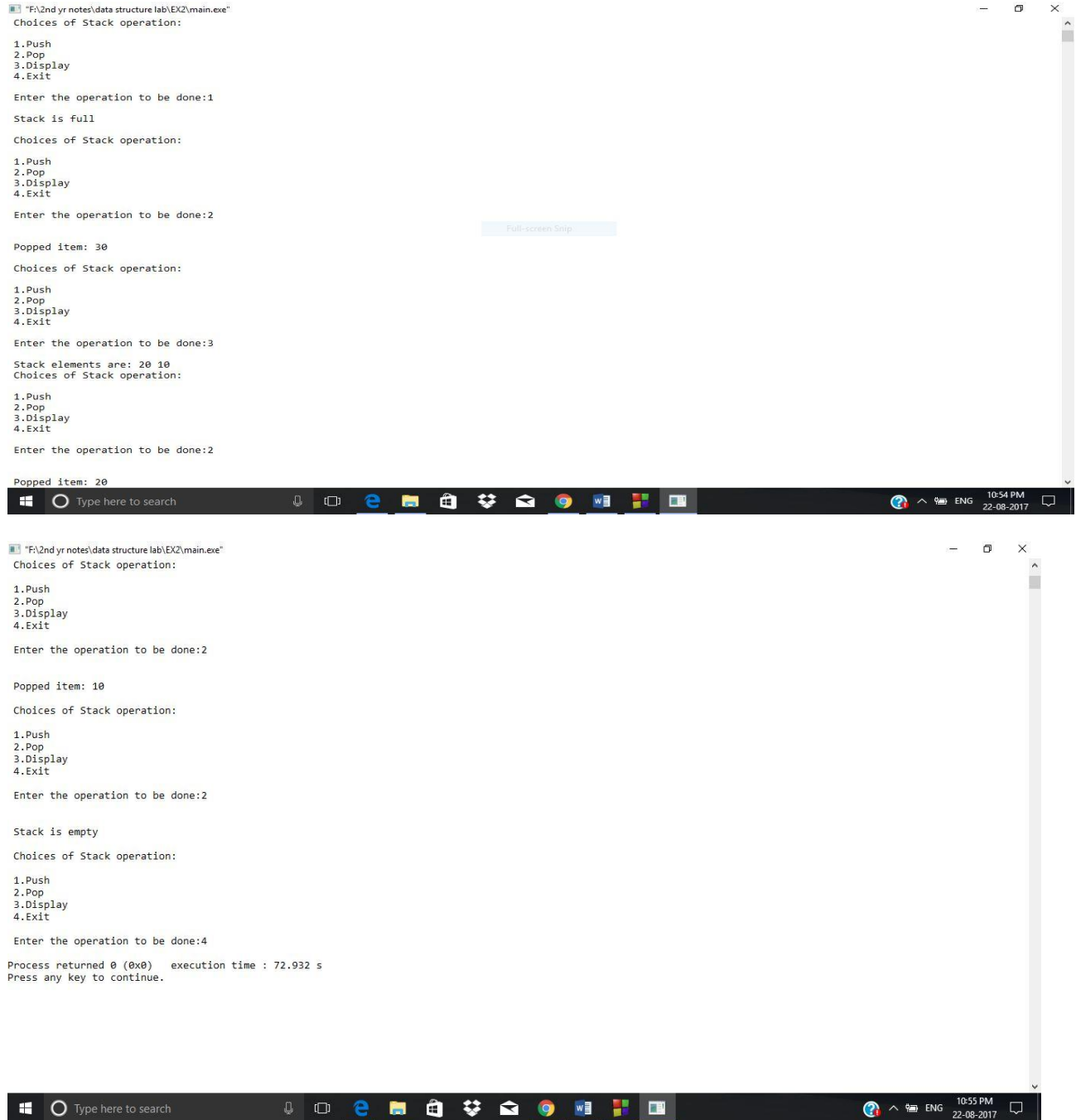


```
main.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
main0: int
Management
Projects Symbols Files
Workspace
Start here main.cpp
52
53
54     else{
55
56         cout<<"\n\n Popped item: "<<stk[top]<<endl;
57         top--;
58     }
59     break;
60 case 3:
61     if(top==0)
62     {
63         cout<<"\n Stack is empty"<<endl;
64     }
65     else
66     {
67         cout<<"\n Stack elements are: ";
68         for(i=top;i>=0;i--)
69             cout<<stk[i]<<" ";
70     }
71     break;
72 default:
73     exit(0);
74 }
75 }
76 }
77 return 0;
78 }
79
80
Logs & others
F:\2nd yr notes\data structure lab\EX2\main.cpp
Windows (CR+LF) WINDOWS-1252 Line 69, Column 32 Insert Read/Write default 10:50 PM 22-08-2017
```



```
"F:\2nd yr notes\data structure lab\EX2\main.exe"
Enter the size of the array: 3
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:1
Enter the element:10
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:1
Enter the element:20
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:1
Enter the element:30
```

DATA STRUCTURES LAB



```
"F:\2nd yr notes\data structure lab\EX2\main.exe"
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:1
Stack is full
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:2
Popped item: 30
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:3
Stack elements are: 20 10
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:2
Popped item: 20
Stack elements are: 10
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:2
Popped item: 10
Stack is empty
Choices of Stack operation:
1.Push
2.Pop
3.Display
4.Exit
Enter the operation to be done:4
Process returned 0 (0x0)   execution time : 72.932 s
Press any key to continue.
```

VIDEO URL:

<https://youtu.be/NFh8ZzKFnRo>

RESULT:

The program of array implementation of stack is implemented successfully and the output is verified.