

# 18CS2007 Data Structures and Algorithms Lab

Ex. No:1.A	<b>STACK</b>
Date:08/08/2020	
<p><b>Aim:</b> To make an array representation of stack in C.</p> <p><b>Description:</b> Perform</p> <ol style="list-style-type: none"> <li>1.Push</li> <li>2.Pop</li> <li>3.Display</li> </ol> <p>Under stack.</p> <p><b>Pseudocode:</b></p> <p><b>1)PUSH</b></p> <ol style="list-style-type: none"> <li>1.Check whether stack is <b>full</b>.</li> </ol> <p><b>TOP=SIZE-1</b></p> <ol style="list-style-type: none"> <li>2.If stack is full, then display “<b>Stack is full</b>” and then terminate the loop.</li> <li>3.If <b>not full</b>, increment the value of top by 1.</li> </ol> <p><b>TOP=TOP+1</b></p> <ol style="list-style-type: none"> <li>4.Assign a value to stack[top]</li> </ol> <p><b>STACK[TOP]=value</b></p> <p><b>2)POP</b></p> <ol style="list-style-type: none"> <li>1.Check whether stack is empty</li> </ol> <p><b>TOP== -1</b></p> <ol style="list-style-type: none"> <li>2. If empty, display “<b>stack is empty, deletion is not possible</b>” and terminate.</li> <li>3.If not empty, then define a variable and initialize it with top.</li> <li>4.Display <b>stack[top]</b>.</li> <li>5.Decrement top by one.</li> </ol> <p><b>Top—</b></p> <p><b>3)DISPLAY</b></p> <ol style="list-style-type: none"> <li>1.Check whether stack is empty</li> </ol> <p><b>TOP== -1</b></p> <ol style="list-style-type: none"> <li>2.If empty, display “<b>stack is empty</b>” and terminate.</li> <li>3.If not empty, initialize a variable and assign it to top and execute the loop until stack is equal to Stop.</li> <li>4.Display stack[i] and decrease the value of I by 1.</li> </ol>	

**Code:**

**Online GDB**

**link:**

<https://www.onlinegdb.com/fork/Syzb3-hWw>

```
#include<stdio.h>
int s[10],top = -1;
void push();
void pop();
void display();
void main()
{
    int choice;
    while(1)
    {
        printf("\n1.Push\n");
        printf("2.Pop\n");
        printf("3.Display\n");
        printf("Enter your choice\t:");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
                printf("\nPush\n");
                push();
                break;
            case 2:
                printf("\nPOP");
                pop();
                break;
            case 3:
                printf("\nDisplay");
                display();
                break;
            default:
                printf("\nwrong choice\n");
        }
    }
}

void push()
{
    int value;
    if(top==9)
```

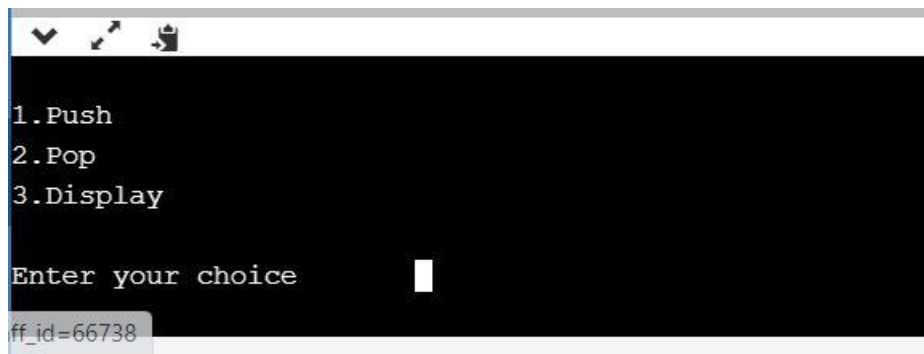
```
{
    printf("Stack is full");

}
else{
    printf("\nEnter the element to push:");
    scanf("%d",&value);
    top=top+1;
    s[top]=value;
}
}

void pop( )
{
    if(top== -1)
    {
        printf("\nStack is empty,deletion is not possible");
    }
    else{
        printf("\ndeleted element is %d",s[top]);
        top=top-1;
    }
}

void display( )
{
    int i;
    if(top== -1)
    {
        printf("\nStack is empty");
    }
    else
    {
        printf("\nstack is..\n");
        for(i=top;i>=0;--i)
            printf("%d\n",s[i]);
    }
}
```

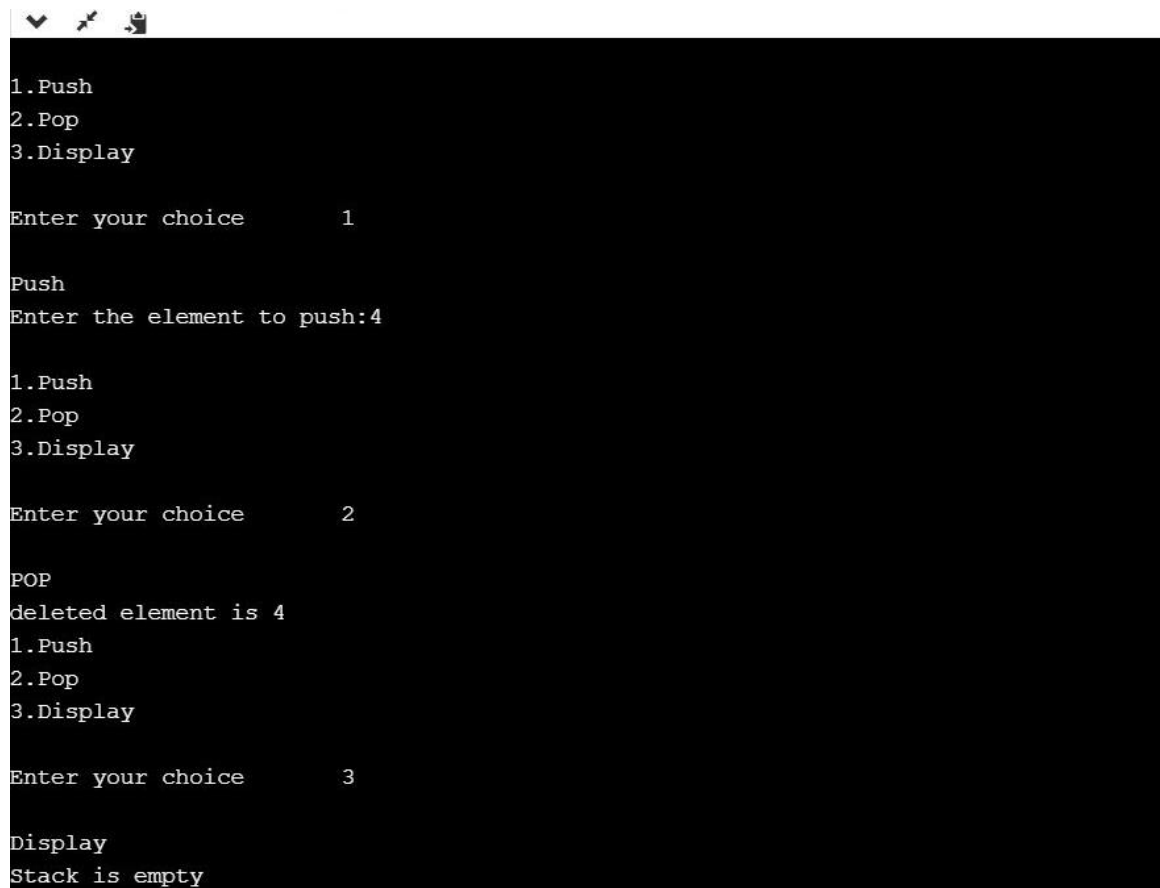
### Sample Input:



```
1.Push
2.Pop
3.Display
Enter your choice
```

A terminal window with a black background and white text. The window has a title bar with three icons (a downward arrow, a magnifying glass, and a trash can). The text inside the terminal shows a menu with three options: 1.Push, 2.Pop, and 3.Display. Below the menu, it says 'Enter your choice' followed by a white cursor. At the bottom left of the terminal, there is a small grey box containing the text 'ff\_id=66738'.

### Sample Output :



```
1.Push
2.Pop
3.Display
Enter your choice      1
Push
Enter the element to push:4
1.Push
2.Pop
3.Display
Enter your choice      2
POP
deleted element is 4
1.Push
2.Pop
3.Display
Enter your choice      3
Display
Stack is empty
```

A terminal window with a black background and white text. The window has a title bar with three icons (a downward arrow, a magnifying glass, and a trash can). The text inside the terminal shows the execution of a stack program. It starts with the same menu as the input: 1.Push, 2.Pop, 3.Display. The user enters '1'. The program prompts 'Push' and then 'Enter the element to push:4'. The menu is shown again. The user enters '2'. The program outputs 'POP' and 'deleted element is 4'. The menu is shown again. The user enters '3'. The program outputs 'Display' and 'Stack is empty'.

## 18CS2007-Data Structures and Algorithms Lab

---

**Result:** Thus the program to perform push ,pop and display is executed and verified successfully.

