18CS2007 Data Structures and Algorithms Lab

Ex. No:1.A	**STACK**
Date:08/08/2020	

Aim: To make an array representation of stack in C.

Description:
Perform
1.Push
2.Pop
3.Display
Under stack.

Pseudocode:

1)PUSH

1.Check whether stack is **full**.

TOP=SIZE-1

2.If stack is full, then display "**Stack is full"** and then terminate the loop.

3.If **not full**, increment the value of top by 1.
TOP=TOP+1

4.Assign a value to stack[top]
STACK[TOP]=value

2)POP

1.Check whether stack is empty
TOP==-1
2. If empty, display "**stack is empty, deletion is not possible"** and terminate.

3.If not empty, then define a variable and initialize it with top.
4.Display **stack[top].**
5.Decrement top by one.
Top—

3)DISPLAY
1.Check whether stack is empty
TOP==-1
2.If empty, display "**stack is empty"** and terminate.
3.If not empty, initialize a variable and assign it to top and execute the loop until stack is equal to Stop.
4.Display stack[i] and decrease the value of I by 1.

Code:

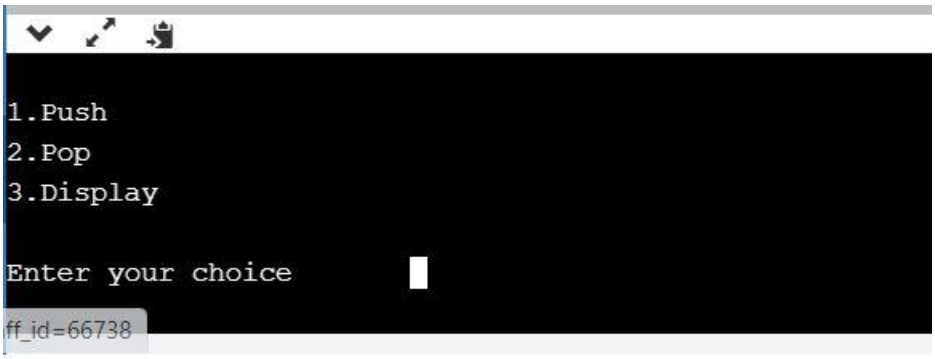
Online GDB link:
https://www.onlinegdb.com/fork/Syzb3-hWw

```
#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)
```

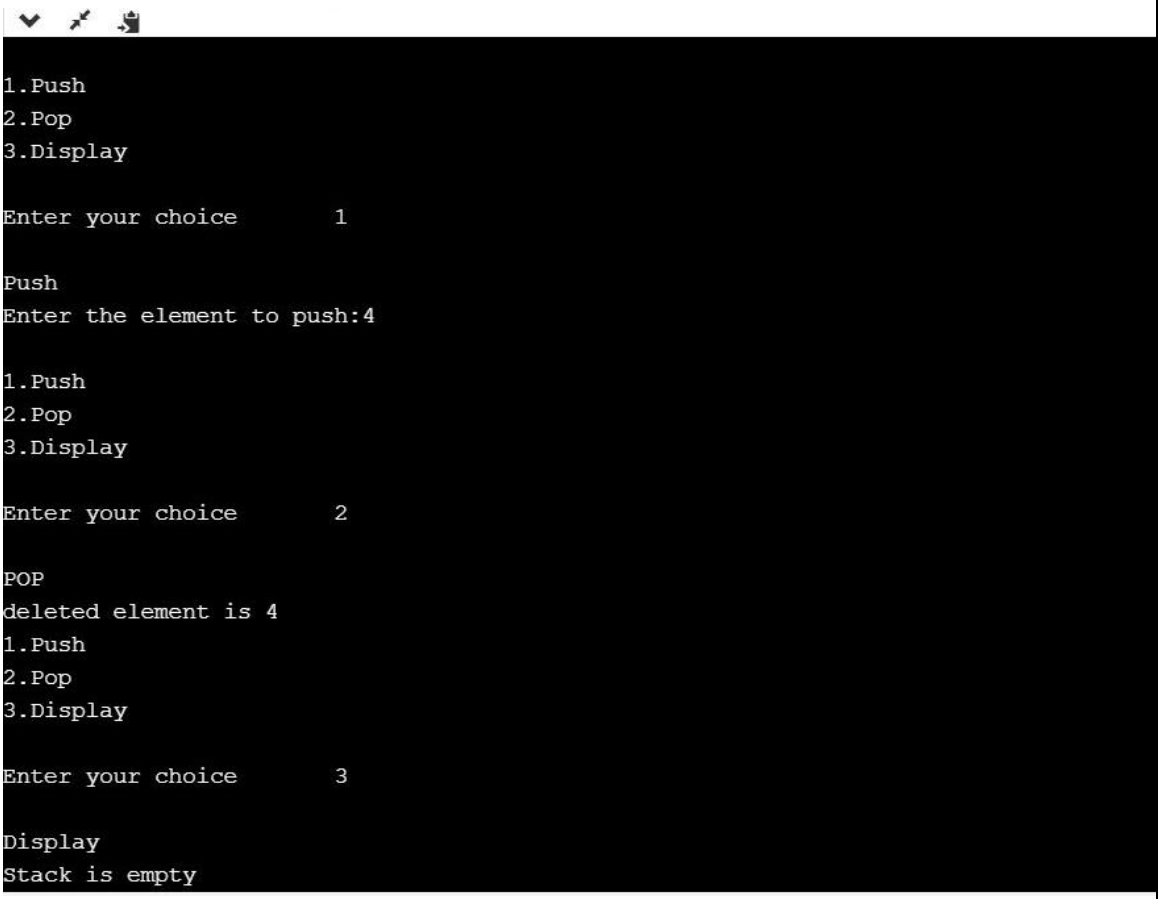
```
    {
       printf("Stack is full");

    }
    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:



Sample Output :



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