

Ex. No.  
29

## Developing and executing 'C' program to implement Stack operation

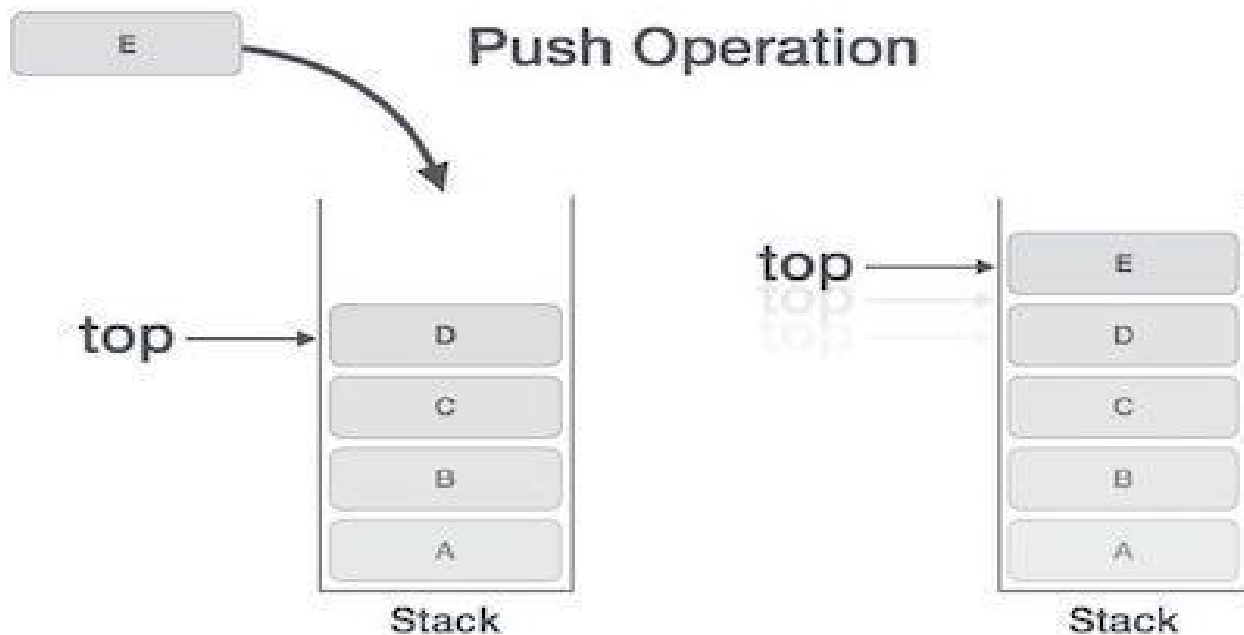
Date :

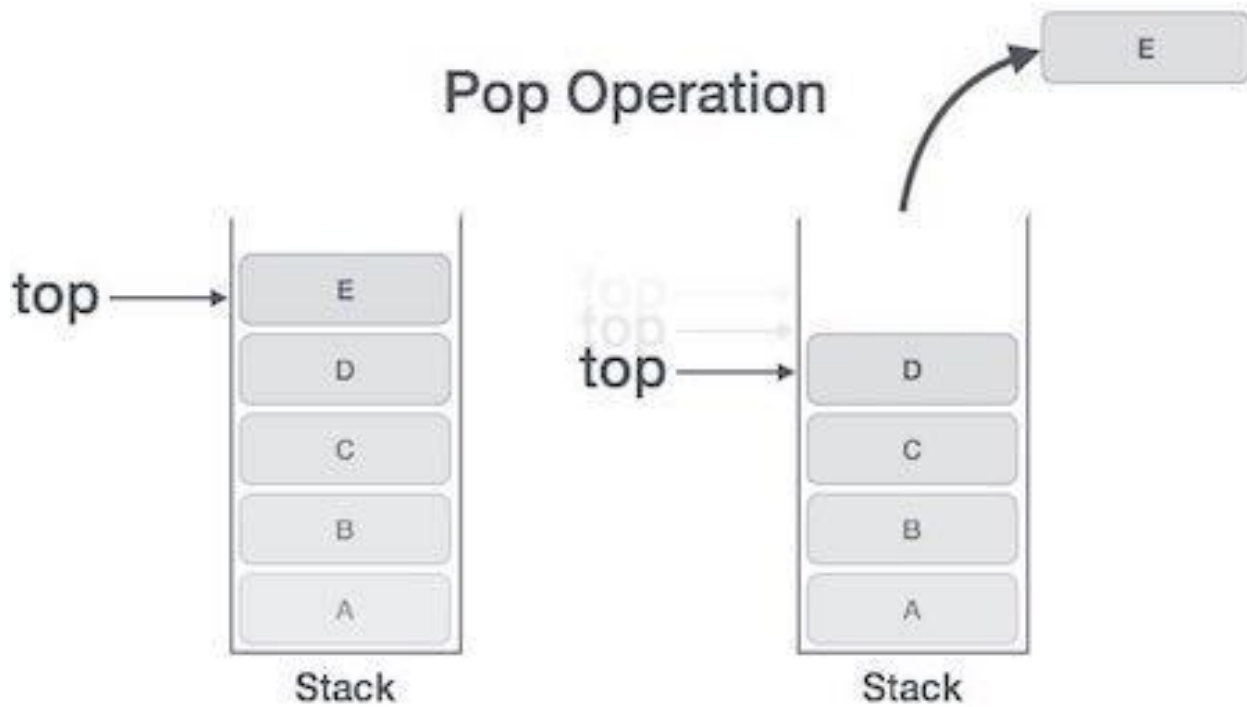
### STACK

- ☐ Stacks are linear data structures in which elements can be inserted and deleted according to a policy decision
- ☐ The element which is inserted last will be the first to be removed
- ☐ This is referred to as **last-in-first-out (LIFO) system**

### Operations Performed On Stack

- ☐ TOP - The end at which the elements are inserted is referred to as TOP of the stack
- ☐ PUSH - The insertion of elements into a stack is referred to as the PUSH operation
- ☐ POP - The deletion of elements from a stack is referred to as the POP operation





**Exercise :** Write a 'C' program to implement Stack operation

```

/*stack program Operation */
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#define max 5
int top= -1,stack[max];
void push( );
void pop( );
void display( );
void main( )
{
    int ch;
    clrscr( );
    while(1)
    {
        printf("\n STACK MENU");
        printf("\n\n 1.Push \n 2.Pop \n 3.Display \n 4.Exit");
        printf("\n ENTER YOUR CHOICE:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:push();
            break;

```

```

        case 2:pop();
            break;

        case 3:display();
            break;

        case 4:exit(0);

        default: printf("\n Wrong choice");
    }/*switch*/
} /*while*/
}/*main*/

/*Function for push operation */
void push( )
{
    int val;
    if(top==max-1)
    {
        printf("\nSTACK IS FULL");
    }
    else
    {
        printf("\nENTER THE ELEMENT TO PUSH:");
        scanf("%d",&val);
        top=top+1;
        stack[top]=val;
    }
}

/*push function */

/*Function for push operation */
void pop( )
{
    if(top==-1)
    {
        printf("\nSTACK IS EMPTY");
    }
    else
    {
        printf("\n DELETED ELEMENT IS %d",stack[top]);
        top=top-1;
    }
}

/*pop function */

/*Function for push operation */
void display( )
{

```

```
int i;
if(top==-1)
{
    printf("\n STACK IS EMPTY");
}
else
{
    printf("\n Stack is\n\n");
    for(i=top;i>=0;--i)
        printf("\n%d",stack[i]);
}
}/*Display function*/
```