Top=4	60
Top=3	15
Top=2	30
Top=1	20
Top=0	10

Top=-1

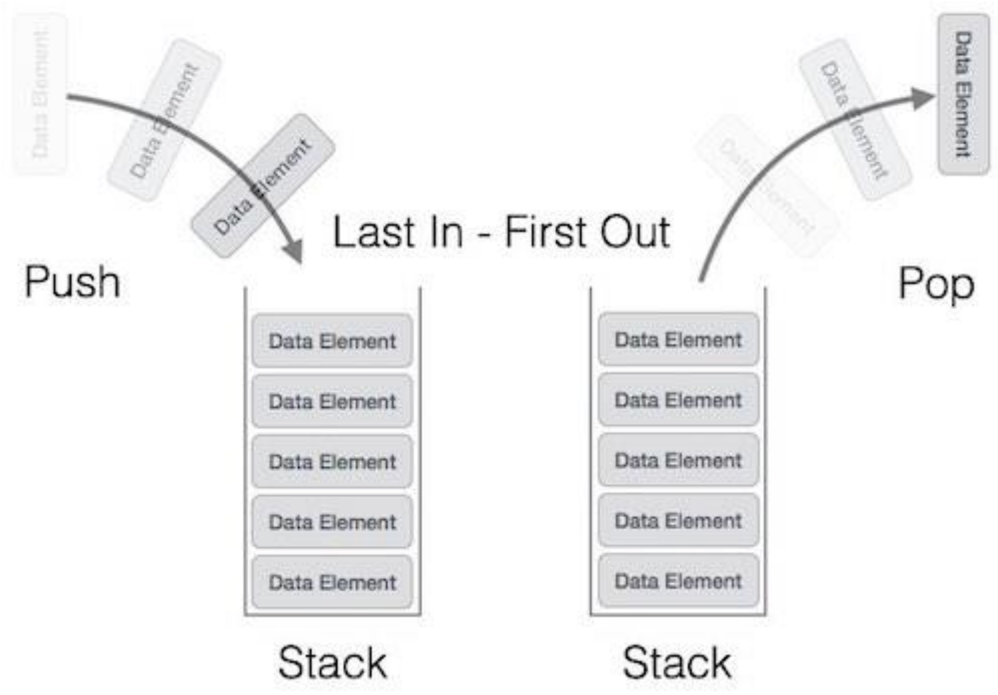
Top++

TOP= Pointer

n=5[0 - 4]

if top=n-1

top=5-1=4



Stack = Last in first out(LIFO)

Stack operations:

1. Push=insert
2. Pop=delete
3. Peep=view
4. Edit=change

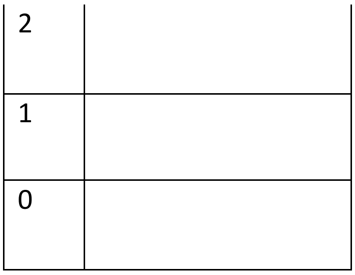
1. **Push (insert)**

```
Step 1. [check stack overflow]
If top = n-1
    Then write “stack overlow”
Return

Step 2. [Increment TOP pointer]
Top=Top+1

Step 3. [insert element]
S[top]=data

Step 4. [finish]
return
```



Top=-1

2. POP(delete)

```
Step 1.[Check for the underflow condition]
If top = -1
    Then write “stack underflow”
Exit
Step 2.[assign top element of stack to data]
```

Data =s[top]

Step 3.[decrement top pointer]

Top = top -1

Step 4. [return top element of stack]

Return data;

Top=4	25
Top=3	15
Top=2	30
Top=1	20
Top=0	24

Top=-1

S[top]-i+1=x   x=25

4-4+1=1

Peep=view particular element

```
Step 1:[ check for the stack underflow]
If top-i+1<0
    Then write (‘stack underflow’)
    Exit
Step2:[return the element from top of stack]
    Return (s[top]-i+1)
Step 3:[finish]
return
```

Edit=change

```
Step 1:[ check for the stack underflow]
If top-i+1<0
    Then write ('stack underflow')
    Exit

Step 2:[change the ith element]
S[top]-i+1=x

Step 3[finish]
return
```