

## ❖ INTRODUCTION TO STACK

### 1. STACK

A stack is an ordered collection of items (elements), in which the removal and addition of stack items can take place only at the top of the stack.

- To add (push) an item to the stack, it must be placed on the top of the stack.
- To remove (pop) an item from the stack, it must be removed from the top of the stack too.

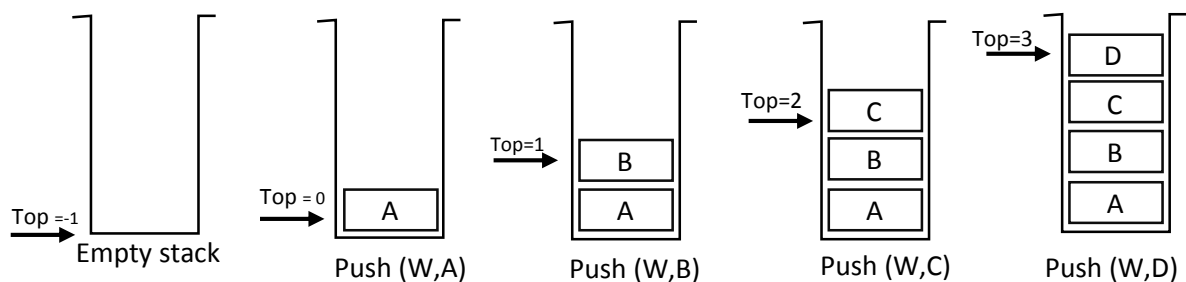
Thus, the last element that is pushed into the stack, is the first element to be popped out of the stack. i.e., Last In First Out (LIFO)

### 2. OPERATIONS OF STACK

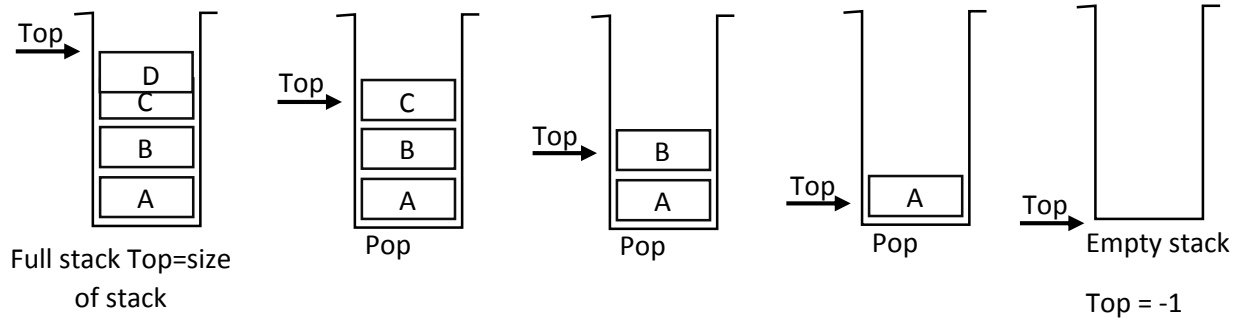
The two main operations which can be applied to a stack are given spatial names

- **Push:** add an element to the top of stack
- **Pop:** delete the element at the top of stack

Example: Performing push operation with item ABCD with a stack W.



Pop operation (delete item from the stack)



### 3. TYPES OF STACKS

- Arrays (static: the size of stack is given initially)
- Linked lists (dynamic: never become full)

### 4. STACK ALGORITHMS

There are two algorithms to add and remove items into and from stack.

#### ➤ PUSH ALGORITHM

##### 1-[over flow]

If  $Top \geq N$

Then over flow

##### 2-[increment pointer]

$Top \leftarrow Top + 1$

##### 3-[insert element]

$Stack[Top] \leftarrow \text{New element}$

## ➤ POP ALGORITHM

1-[under flow]

    If  $Top \leq -1$

        Then under flow

2-[unstack element ]

$element \leftarrow Stack[Top]$

3-[decrement pointer]

$Top \leftarrow Top - 1$

Notes: Top indicate to the top of stack which initialized (top = -1)

## 5-DECLARATION OF STACK

```
Struct    stack
```

```
{ int  top ;
```

```
int  item [stacksize] ;
```

```
};
```

### // RUN OPERATION OF STACK

```
#include<iostream.h>
```

```
struct stack
```

```
{
```

```
int top;
```

```
int  item [10] ;
```

```
};
```

```
main()
```

```
{
```

```
stack s1;
```

```
s1.top=-1;
```

```
char c; int item;
```

```
// push to stack
```

```
cout<<" \n Do  you want to Push in to stack (y,n): ";
```

```
cin>>c;
```

```
while(c=='y')
```

```
{
```

```
if(s1.top>=10)
```

```
{cout<<"\n error...the stack is full"<<endl; break;}
else
{ cout<<"\n enter item: ";
cin>>item;
s1.top++;
s1.item [s1.top]=item;
}
cout<<" \n Do you want to Push in to stack (y,n): ";
cin>>c;
}
//pop from stack
cout<<" \n Do you want to Pop from stack (y,n): ";
cin >>c;
while (c=='y')
{
if(s1.top<0)
{ cout<<"\n error...the stack is empty \n"<<endl;break; }
else
{
item=s1.item[s1.top];
s1.top-=1;
}
cout<<item<<" :is popped\n ";
cout<<" \n Do you want to Pop from stack (y,n): ";
cin >>c;
}
}
```