Stack

* What is Stack?

↳ Stack is an order List in which Elements are to be inserted and deleted From one end called top end. Stack is LIFO: Last in first Out.

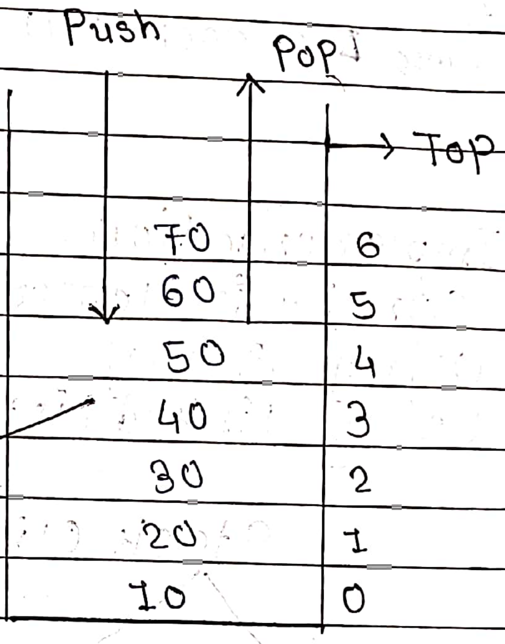
In other word.

- A stack is a linear data structure that follows the LIFO Principle Last in First out.
The last element added is the first one to be removed.

Date :
P. No. :

* Stack Operations. (Basic)

Push : Insert an element
Pop :- Remove the top element
Display :- Show all elements of the stack.

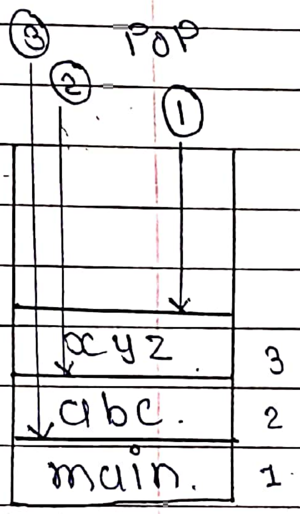


Implementation
- Array
- Linked List.

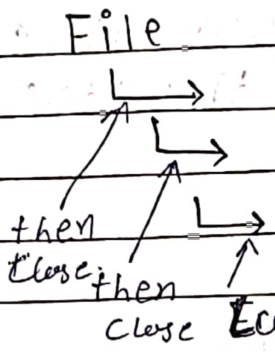
* Application of Stack. (Technical).

- Function Call Handling
↳ Ex.

```
① { void main ()
    {
        → abc ();
    }

② { abc ()
    {
     → xyz ();
    }

③ { xyz ()
    {
        =
    }
```

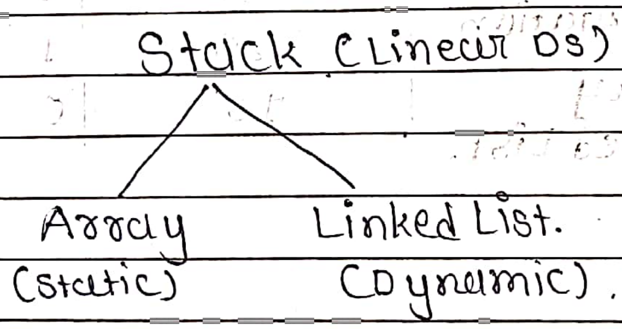


- Rucursion.
- Menu Driven.
  ↳ Ex.



Last open But First Close

* Stack Implementation.
- Stack can be implemented in two ways:
  - Using Array
  - Using Linked List.



Date :
P. No. :

- Implementation Stack using Array.

```
#include <stdio.h>
#define Maxsize 10

int Stack[Maxsize], Top=-1;

~~void~~ int main ()
    {   int choice;
        do{
         printf("----- Stack -----");
         printf("\n 1. Push \n 2. Pop \n
                 3. Display \n 4. Exit");
         printf("\n ---------------");

         printf("\n Enter Choice : ");
         scanf("%d", &choice);

        switch (choice)
        {
            case 1 : Push(); break;
            case 2 : Pop() ; break;
            case 3 : Display; break;
            default :
                    printf ("\n Invalid Input");
                    break;
        }
        while (choice!=4);
    }
    return 0;
}
```

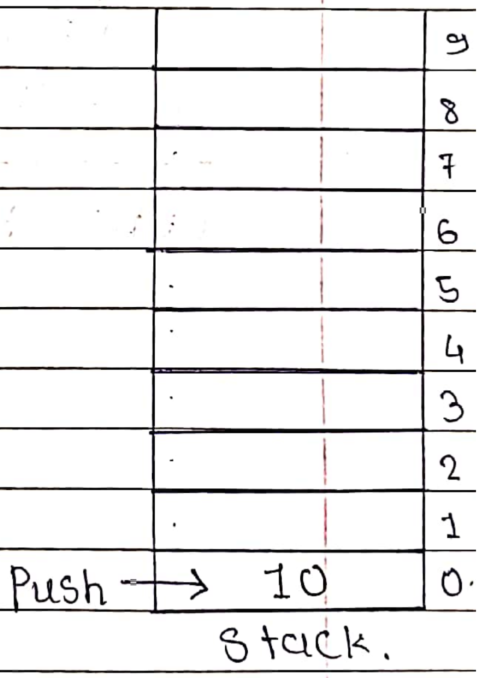
Date :
P. No. :

```
P.void Push ()
 {
      int n;
    if (Top==MaxSize-1)
    {
       printf ("stack is overflow");
    }
    else
    {
        printf("\n Enter Element :");
         scanf ("%d", &n);

         Top ++;

          Stack[Top] = n;
     }
 }

 void Pop ()
 {
      if (Top == -1)
      {
         printf ("Stack is Underflow");
      }
      else
      {
           Top--;
       }
 }
```

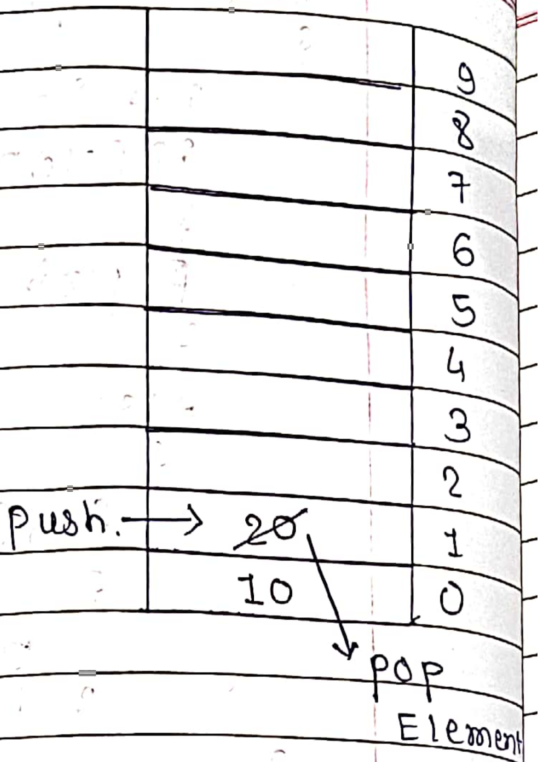
```
void Display()
{
    int i;
    if(Top==-1);
    {
    printf("Stack is Underflow");
    }
    else
     {
       for(i=Top; i>=0; i--)
       {
           printf("%d\n", Stack[i]);
       }
  }
```

=> output.

```
----- Stack -----
  1. Push
  2. Pop
  3. Display
  4. Exit
-------------------
Enter Choice : 1
Enter Element : 10.

----- Stack -----
  1. Push
  2. Pop
  3. Display
  4. Exit
-------------------
```



```
Enter choice : 1.
Enter Element : 20
----- Stack -----
1. Push
2. Pop
3. Display
4. Exit
-----------------
Enter Choice : 2
----- Stack -----
1. Push
2. Pop
3. Display
4. Exit
-----------------
Enter Choice : 3
10
-----Stack-----
1. Push
2. Pop
3. Display
4. Exit.
-----------------
Enter Choice : 4.
```


9
8
7
6
5
4
3
2
push
~~20~~
1
10
0
pop
Element