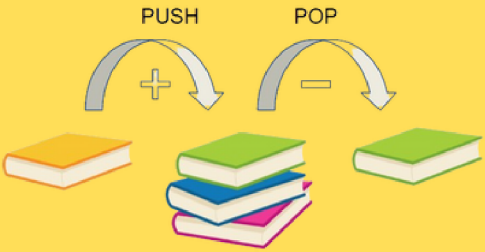
Follow

Data Structure & algorithm

Recursion-2

#Vale_freeContent



Call Stack

SWIPE

Call stack

Whenever we execute any line of code / function in begining, call stack is initialize.

Call stack

↓

A place / stack kin data structure where execution happen.

↓ follows

* FILO :- first in last out order
* LIFO :- Last in First out order.

FILO / LIFO

The thing we put in stack 1st → taken out in the last [FILO]

The thing put in last → taken out in first order [LIFO]

ex → Restaurent

this person pick up the 6th plate first

↳ follow

FILO / LIFO

Push / PoP

When we try to insert anything called → Push

When we try to remove anything from stack called → PoP

⇒ Code

```
f1()
f2()
f3()
```

Create

after 100% of execution → POP $f_1()$

Same things follow $f_2()$, $f_3()$

Once all the execution done → Call Stack should be Empty

Call stack

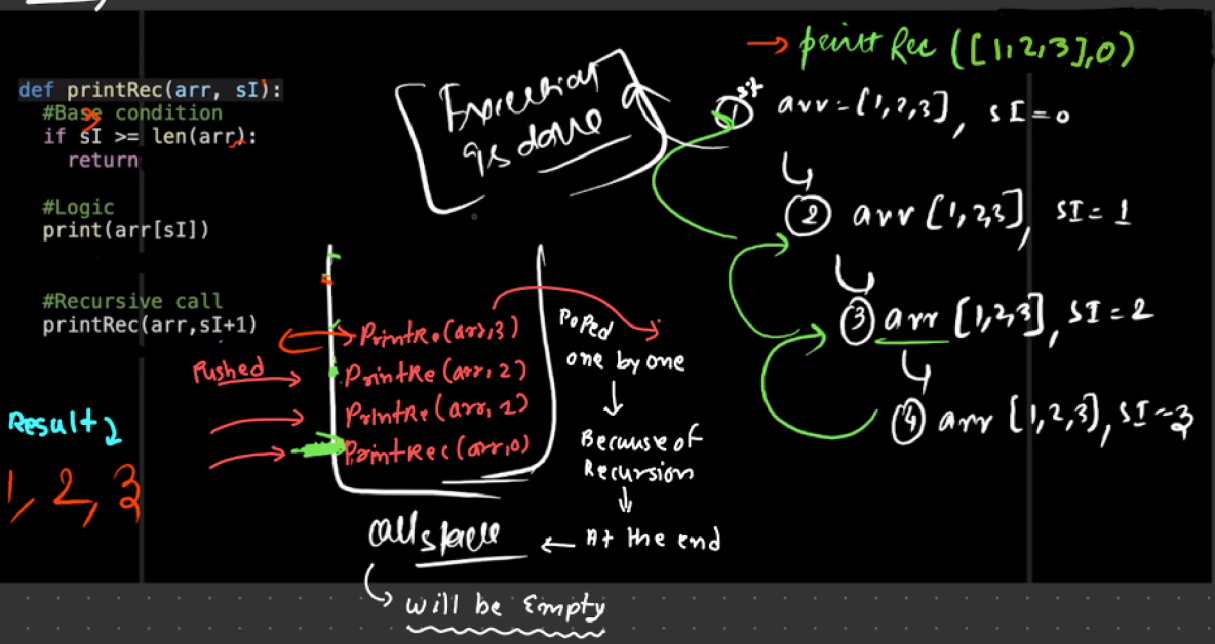
- Empty in Beginning
- Empty in End

ex →

```
def printRec(arr, sI):
  #Base condition
  if sI >= len(arr):
    return

  #Logic
  print(arr[sI])


  #Recursive call
  printRec(arr,sI+1)
```



Q arr = [3, 5, 1, 2, 4], find the minimum Value in this array using Recursion Method.

↳

1st [define a function]

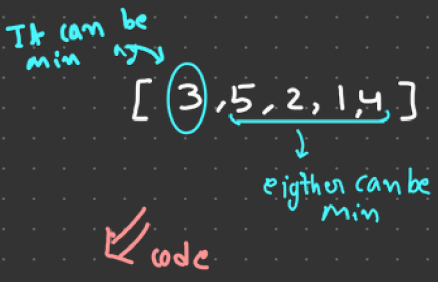
↳ ① what will the name
↳ ② what will the defination of the function.

⇓

↳ [3, 5, 2, 1, 4]

⇒ def findMin(arr, sI):

```
    # Base condition
    if sI >= len(arr):
        return ggggg
```



return min(arr[sI], FindMin(arr, sI+1))

Left part minimum

Right Part as minimum

```python
# Ans

def find_min(arr, sI):
    #Base condition
    if sI >= len(arr):
        return 999999      ## here 99999 is considering large numbder

    #logic
    return min(arr[sI], find_min(arr,sI+1))

#driver code
print("The Smallest value is:", find_min([3,5,1,2,4],0))
```

```
The Smallest value is: 1
```

Q.3 arr = [3,34,543,32,33,76,34,6] find the largest value by using recursion method.

```python
def find_max(arr, sI):

    #Base condition
    if sI >= len(arr):
        return sI

    #Logic
    return max(arr[sI], find_max(arr, sI+1))

# Driver code
print("The Largest value is:", find_max([3,34,543,32,33,76,34,6],0))
```

```
The Largest value is: 543
```