

# CS & IT ENGINEERING

Data Structure

Stack and Queue  
Chapter- 4

Lec- 01



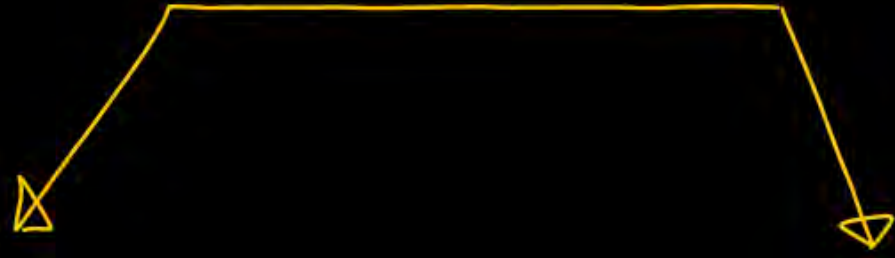
By- Pankaj Sharma sir



TOPICS TO BE  
COVERED

Stack-I

# Data structure



## ① Abstract view

- \* No coding
- \* No implementation
- \* No prog. language

features/operations defined  
on a data structure.

## ② Concrete view

- \* Implementation
- \* prog. language

# Stack

- \* Linear data structure
- \* Deletion order  $\Rightarrow$  reverse order of insertion.
- \* Works on Last-In First Out (LIFO)
- \* Both Insertion and deletion are performed only at one end called TOP of stack.



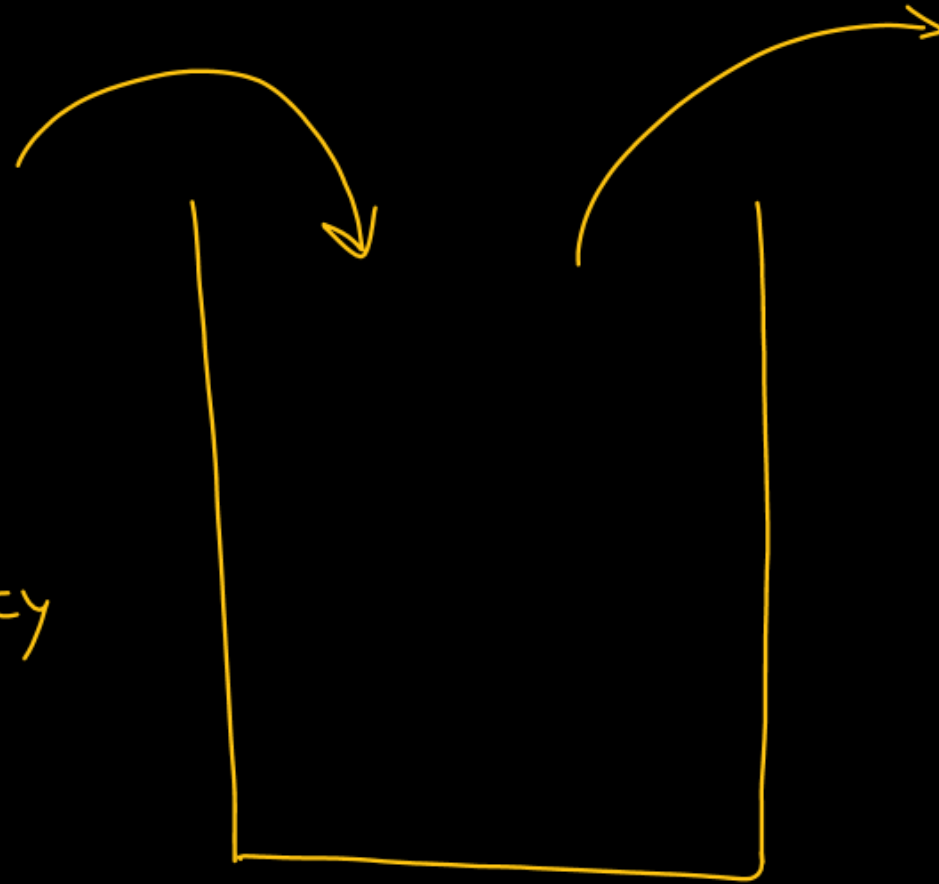
## Stack as an ADT

Stack of numbers  
(Integer)

Initially  $\Rightarrow$  Empty

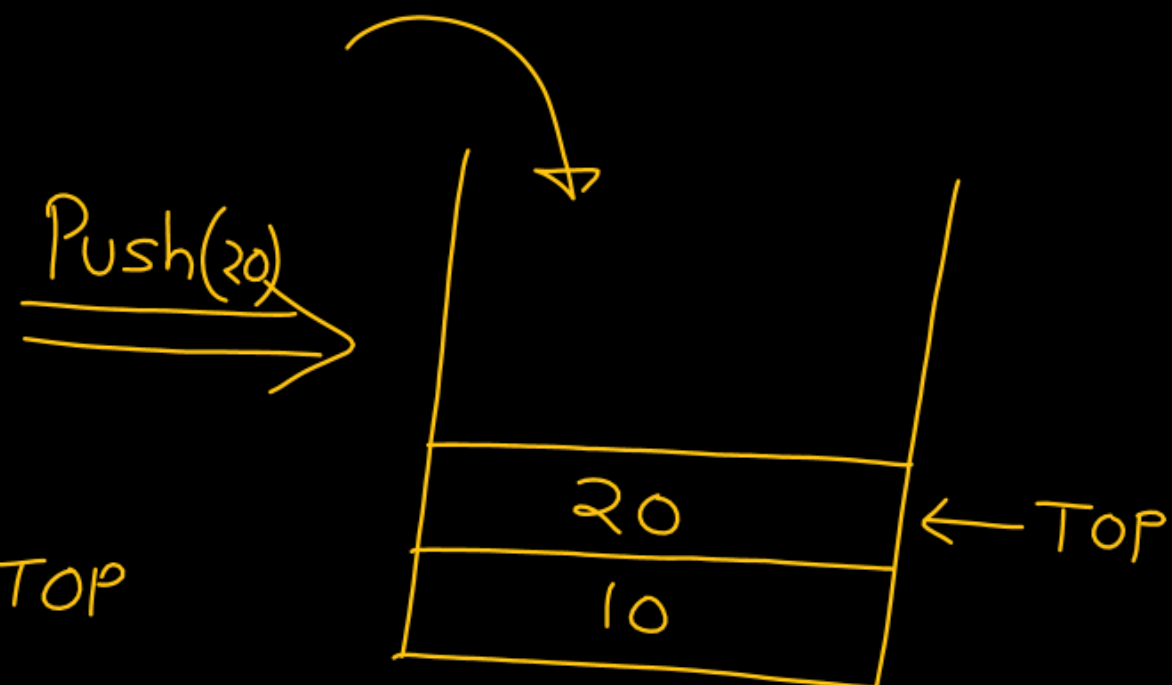
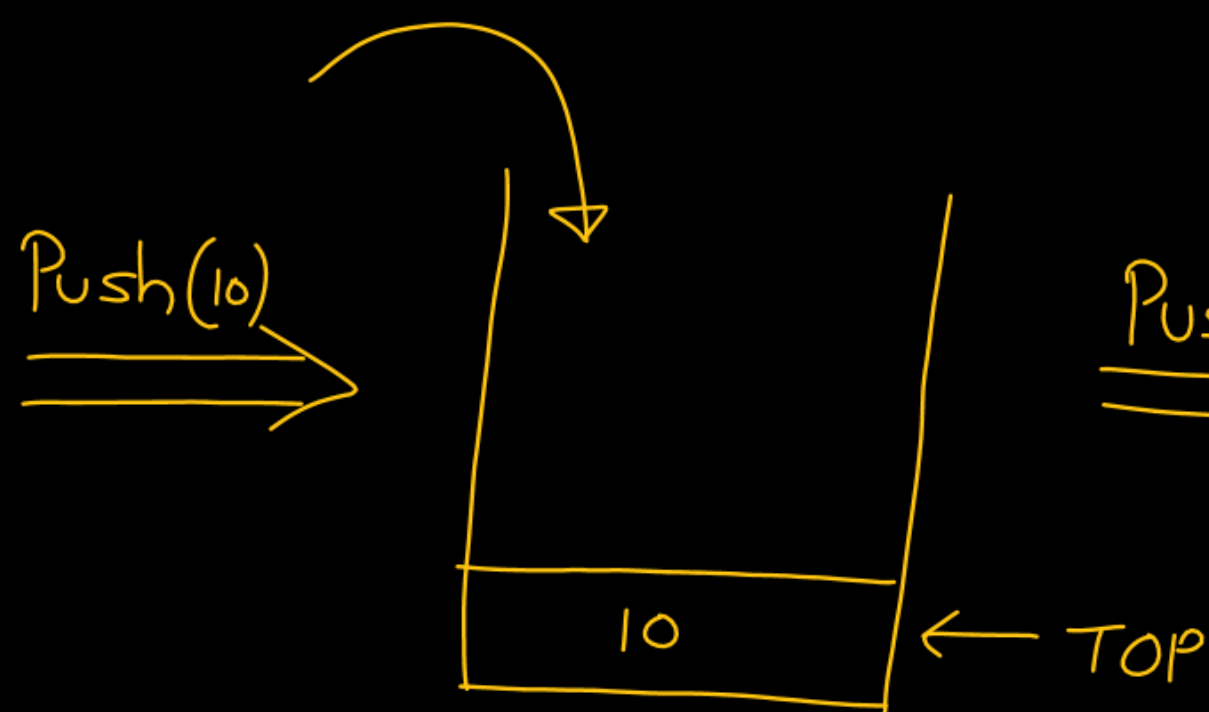
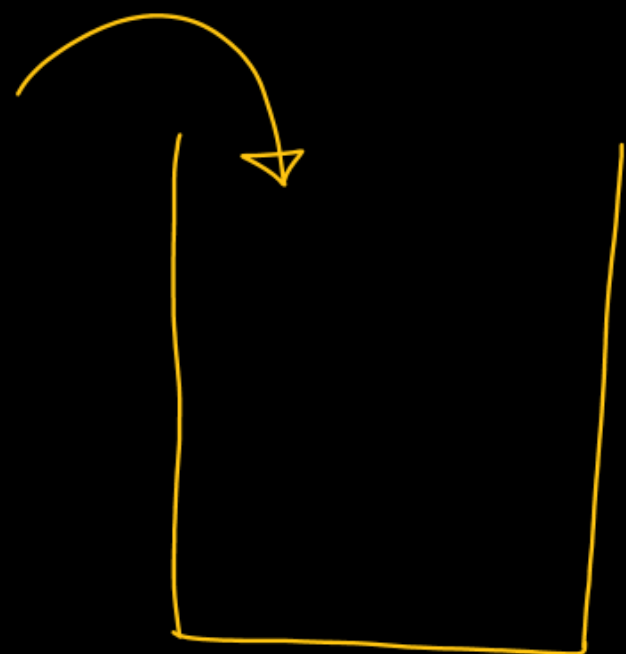


Points to  
most recently added  
element.

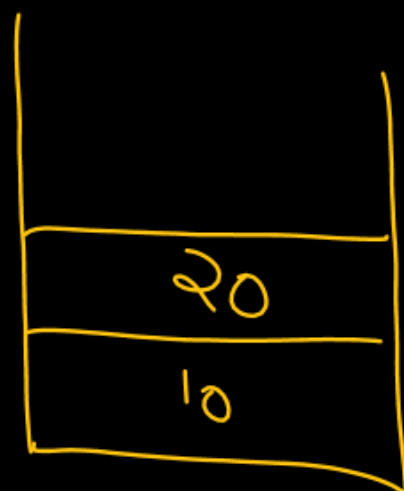




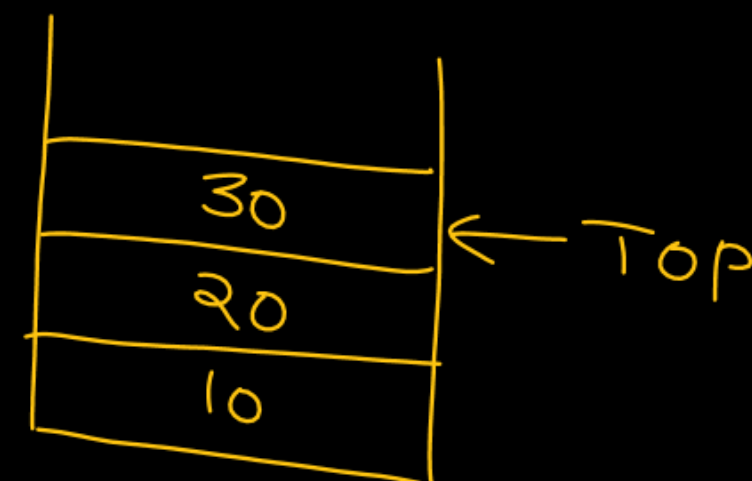
- ① Insert  $\Rightarrow$  Push
- ② Delete  $\Rightarrow$  Pop



False  
IsEmpty()



Pop



IsEmpty() : → true/false

IsFull() → true/false

## Applications

- 1] TOH
- 2] DFS
- 3] infix to postfix
- 4] infix to prefix
- 5] postfix evaluation
- 6] Recursion
- 7] Parenthesis checking

wait ch29/ottl  
to delay ✓

to postponed decision

UNDO







C

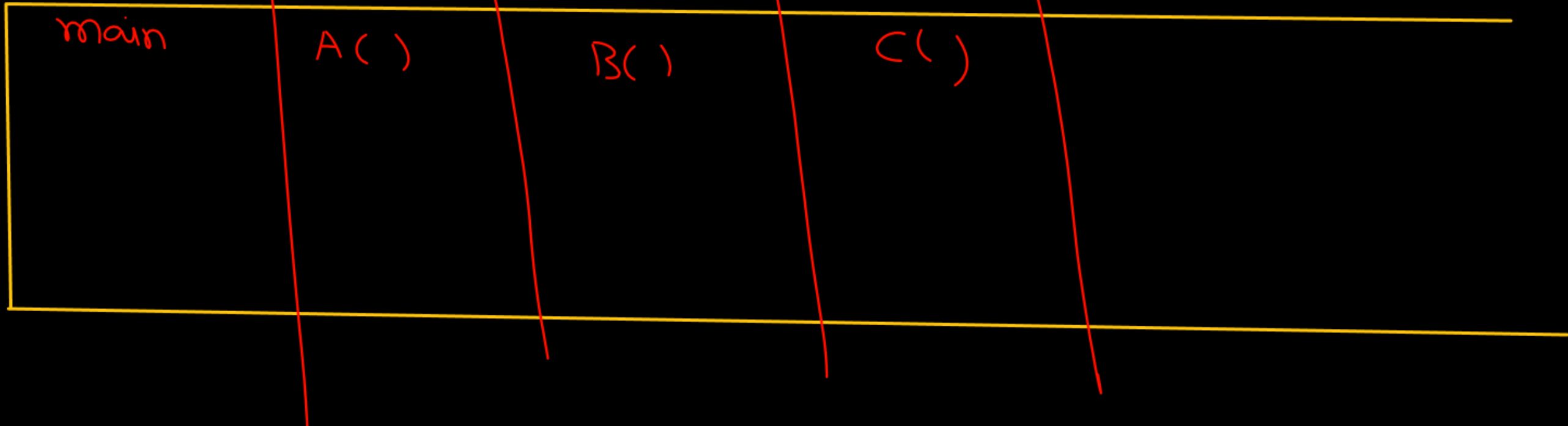
```
main() {  
    ==  
    A()  
    ==  
}
```

```
A() {  
    ==  
    B()  
    ==  
}
```

```
B() {  
    ==  
    C()  
    ==  
}
```

```
C() {  
    ==  
    ==  
    ==  
    ==  
    ==  
}
```

System  
Stack

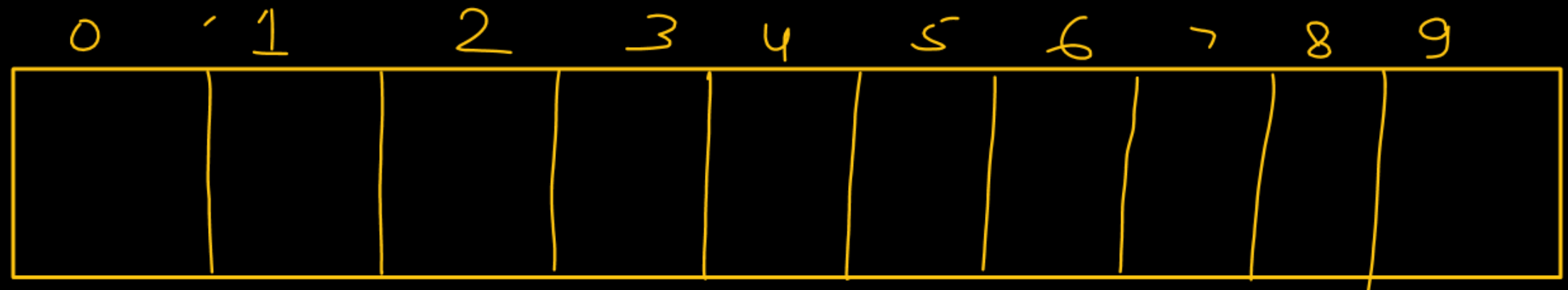


# Stack

① Array : → size fixed  
static

```
#define SIZE 10  
int STACK[SIZE];
```

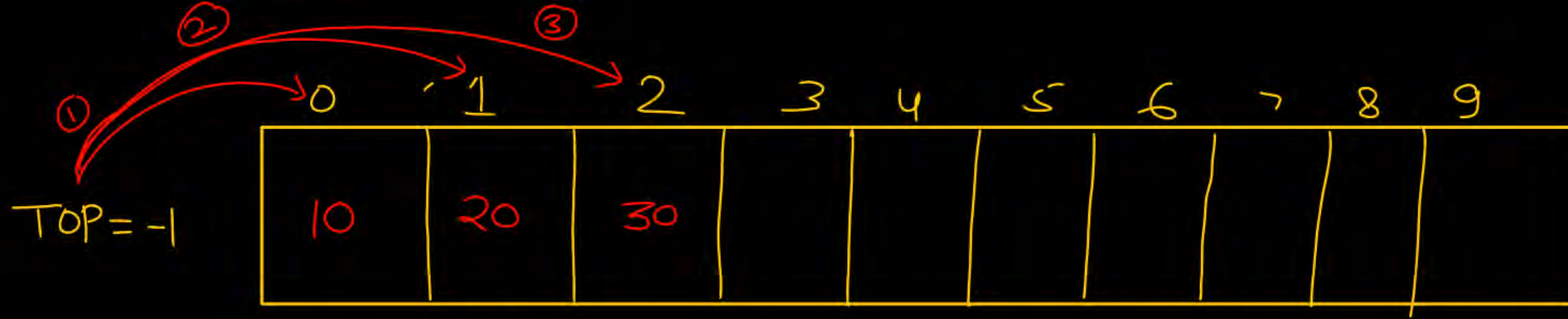
Most recently added  
element



TOP :

Initially

TOP = -1



Top  
-1

1) Push(10)

2) Push(20)

3) Push(30)

4) Push(40)

⋮

9) Push(90)

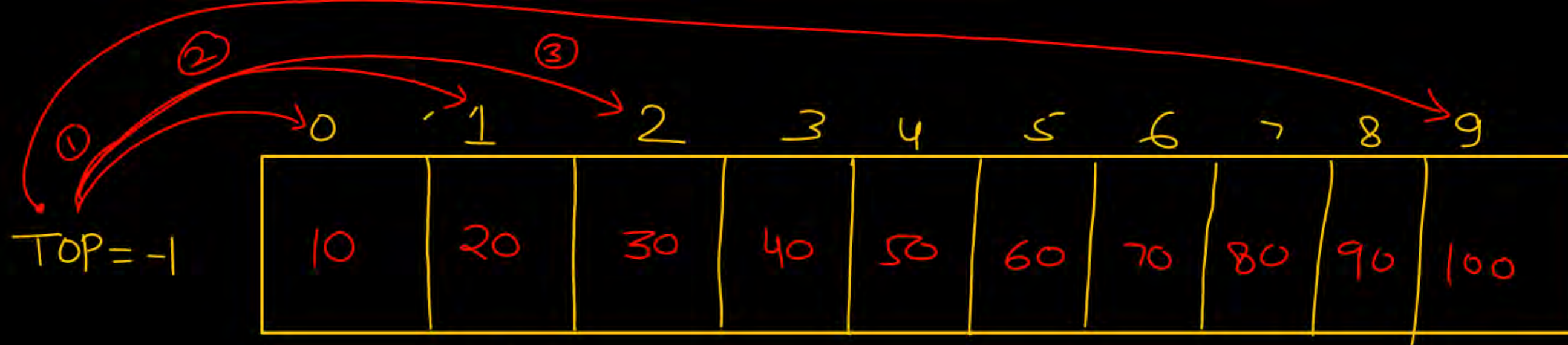
10) Push(100)

↓  
3  
⇒

```
void Push( int x )
{
```

```
    TOP = TOP + 1;
    STACK[TOP] = x;
}
```





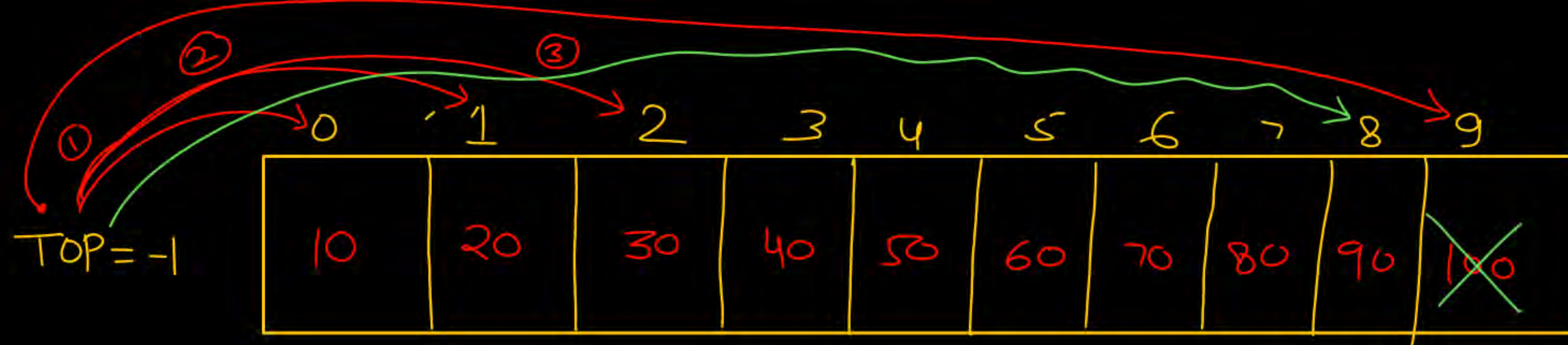
	Top
1) Push(10)	-1
2) Push(20)	0
3) Push(30)	1
4) Push(40)	2
⋮	⋮
9) Push(90)	8
10) Push(100)	9

```

void Push(int x) {
    if (TOP == SIZE - 1)
    {
        constant printf("overflow");
        return;
    }
    TOP = TOP + 1;
    STACK[TOP] = x;
}
  
```

SIZE 10  
 last valid index  
 = 9  
 (SIZE - 1)

OR  
 $STACK[++TOP] = x;$



10, 20, 30, ... 70, 80, 90 <sup>X</sup>

Pop : return element

temp = A[9]

int Pop( ) {

int temp;

temp = STACK[TOP];

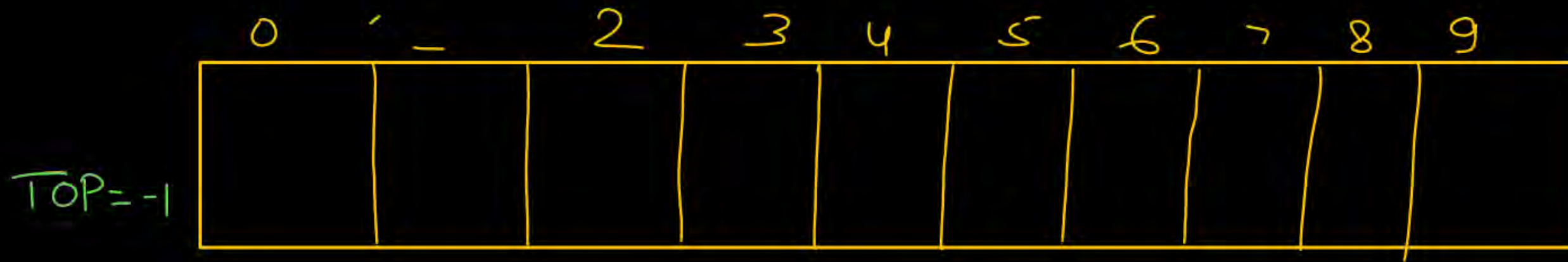
TOP = TOP - 1

return temp;

}

दिल्ट ई





```
int Pop() { int temp;  
    if (TOP == -1) // underflow  
    {  
        return INT_MIN;  
    }  
    temp = STACK[TOP];  
    TOP = TOP - 1;  
    return temp;  
}
```

Constant

```
#define SIZE 10
```

```
int STACK[SIZE];
```

```
int TOP = -1;
```

```
void Push(int x){
```

```
    if (TOP == SIZE - 1)
```

```
        return;
```

```
    TOP++;
```

```
    STACK[TOP] = x;
```

```
}
```

```
int Pop() {
```

```
    temp = STACK[TOP];
```

```
    TOP--;
```

```
}
```

Problem?



```
int TOP = -1;
```

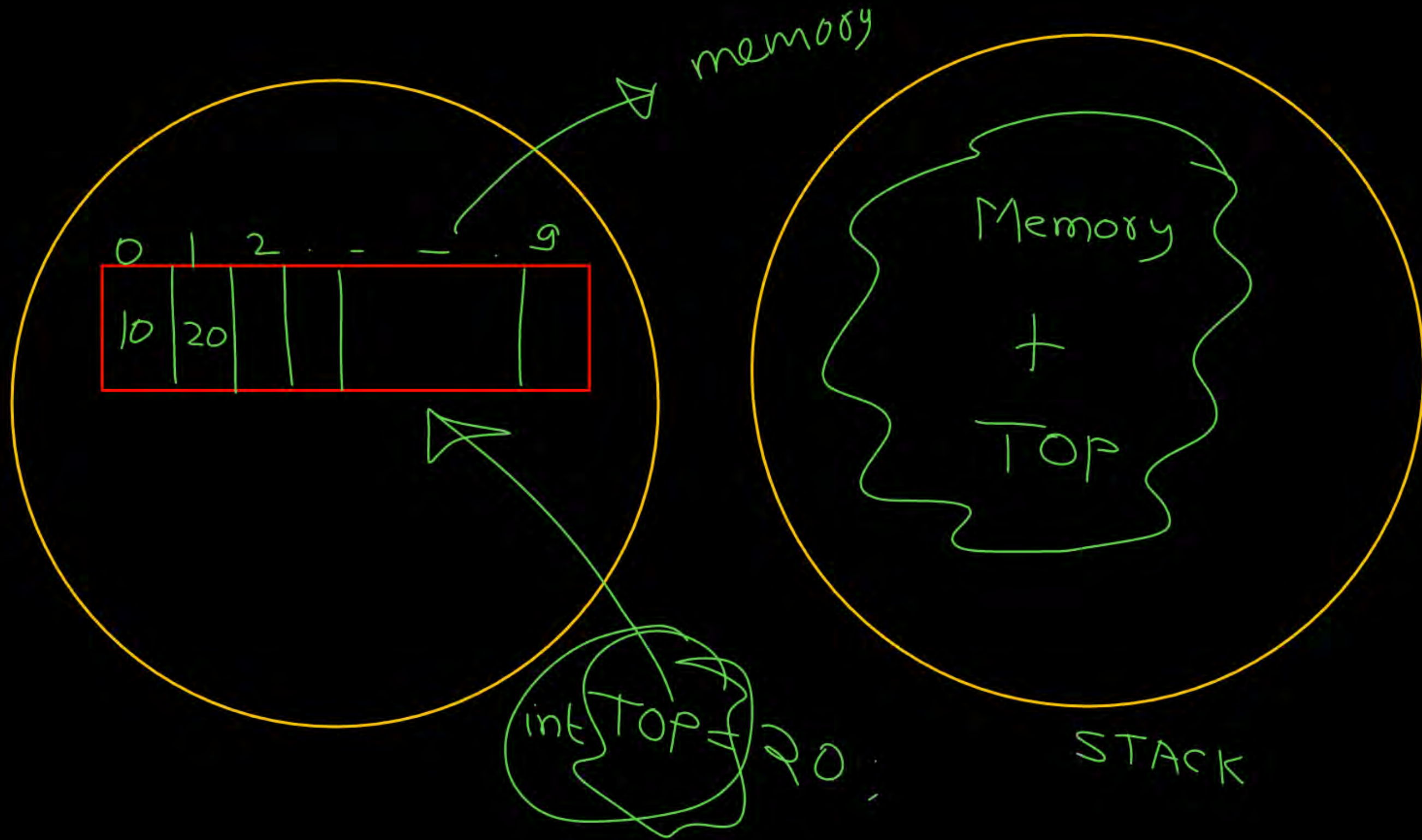
```
int STACK[SIZE];
```

```
void main(){
```

```
    /  
    /  
    /  
    /  
    /
```

```
}
```





#define SIZE 10

struct STACK {

int Array[SIZE];

int TOP;

};

void main() {

struct STACK S;

S.TOP = -1;

==

}

No memory  
is  
allocated

