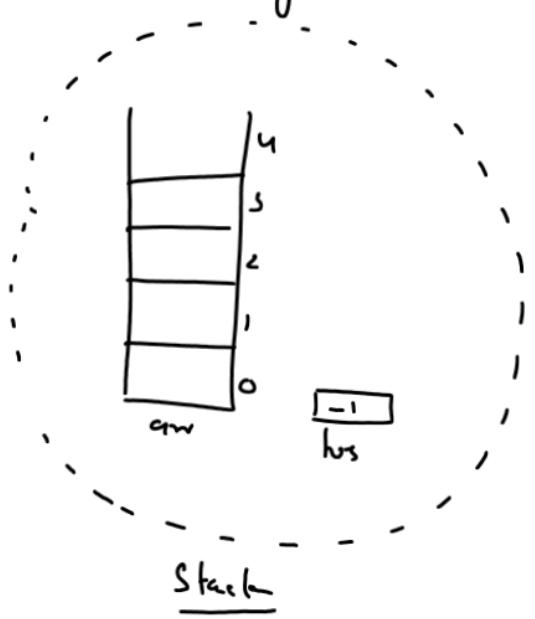
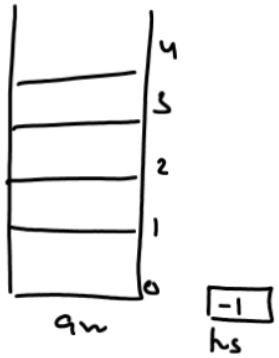
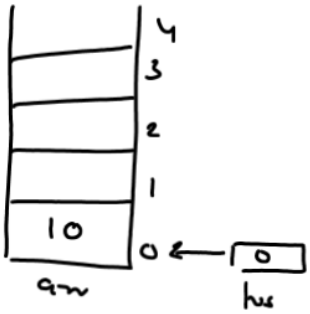
Diagrammatic View of a Stack



① An Empty Stack



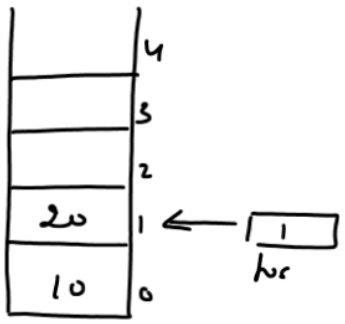
② After inserting 1 ele



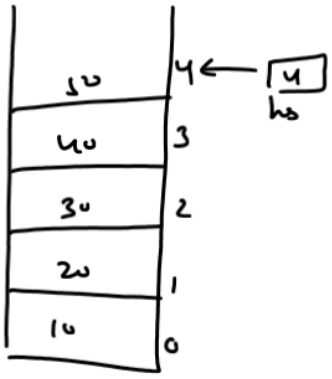
Stack with 2 ele

Full Stack

③



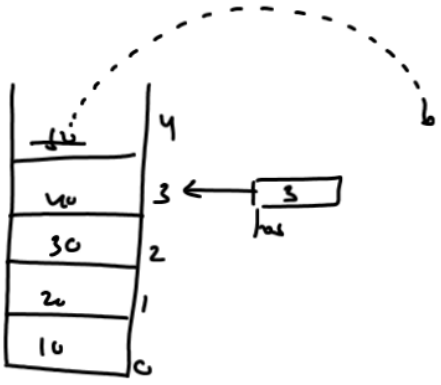
④



⑤ push (60)

"Stack Overflow"

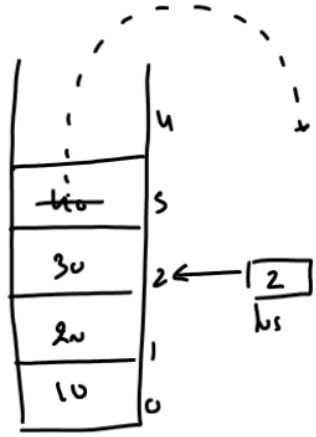
①



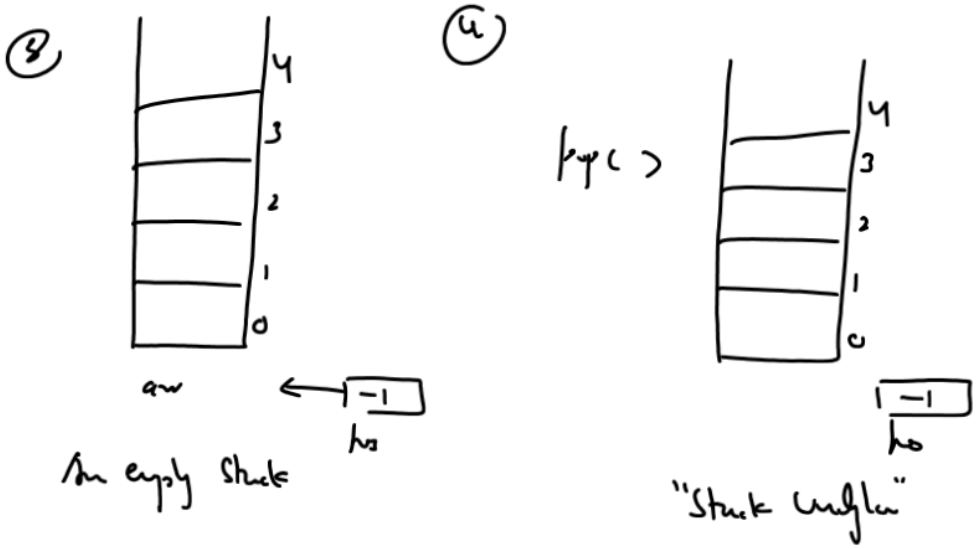
pop()

After popping first ele

②



After popping 2nd ele



Writing Algorithm For push()
==========

What is an algorithm?

An algorithm is STEP BY STEP instructions to solve a particular problem

Algorithms are always language independent i.e they do not user any language specific syntax

People also call algorithm as PSEUDOCODE.

Writng Algorithm For The Function push()
==========

1. Check for overflow.

2. If Stack is full then display STACK OVERFLOW and return otherwise goto step 3.

3. Increment TOS by 1.

4. Insert the element in the STACK at the position pointed by TOS.

5. Return

Writng Algorithm For The Function pop()
==========

1. Check for underflow.

2. If Stack is empty then display STACK UNDERFLOW and return otherwise goto step 3.

3. Delete the element from the STACK from the position pointed by TOS.
4. Decrement TOS by 1.

5. Return the deleted element

Implementing Stack In C

```
struct Stack
{
    int arr[5];
    int tos;
};
```

Qn Why we have used structure for implementing Stack ?

An: Since a Stack is a combination of 2 elements : an array and a tos and both these elements always work together in stack operations so it is a **good programming practice** to declare them together.
Thus we have used a STRCUTURE.