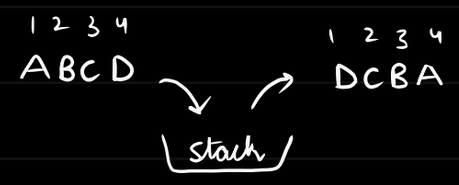
Stack

→ Non primitive data structure.
→ Follows LIFO
→ Insertion & Deletion from only one end.
→ Can access elements only via 'top'.



Implementation :-

→ STL : <1_1_StackUsingSTL.c++>

→ Using Arrays :

Requirements : 1. Stack's max size
2. Array
3. Top index

Operations 1. push : top++
2. pop : top--

} <1_2_StackUsingArray.c++>

→ Using linked list :

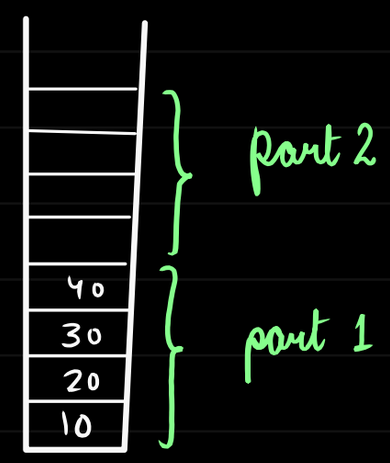
Insert At Beginning ⇒ push()
Delete from Beginning ⇒ pop()

} <1_3_StackUsingLL.c++>

---

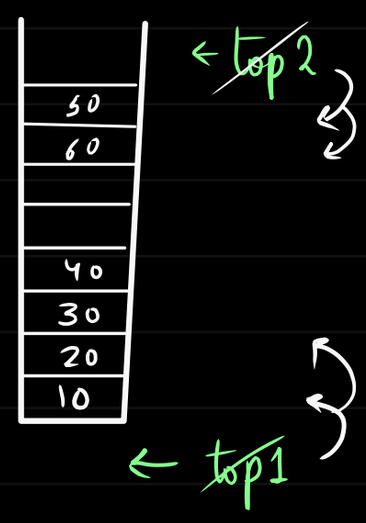
2 Stacks in 1 Array :-

Approach 1 : Divide Array in 2 halves
Cons - Memory wastage
Despite of space in 1 array, other might overflow



Approach 2 : Using 2 indices as 2 pointers.
top 2 : At Array size ⇝ top2--
top1 : At -1 ⇝ top1++

} <1_4_2Stack1Array.c++>



Insertion at bottom of Stack :-
Using Recursion

} <1_6_InsertAtBottom.c++>