Stack

A stack is a linear data structure in which insertion & deletion are allowed only at the end, called the top of the stack.

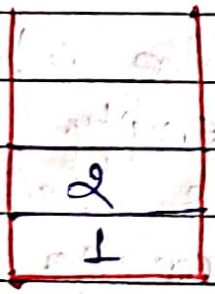


fig:- stack

LIFO → Last In First out.

A stack can be implemented using an array or a linked list.

Application of Stack

1. used in function calls.
2. Infix to postfix conversion (& other similar conversion)
3. parenthesis matching & more.

Stack ADT (Abstract data type)

In order to create a stack we need a pointer to the topmost element along with other element which are stored inside the stack.

Some of the operation of Stack ADT are:

(i) push():→ push an element into the stack.
(ii) pop():→ Remove the topmost element from the stack.
(iii) peek(index):→ value at a given position is returned.
(iv) isEmpty/isFull:→ Determine whether the stack is empty or full.
(v) top():→ returns the last inserted element without removing it.
(vi) size():→ returns the size or the no. of elements in the stack.

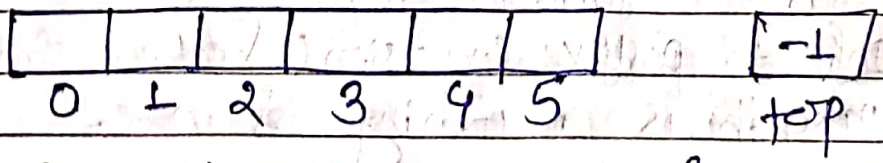
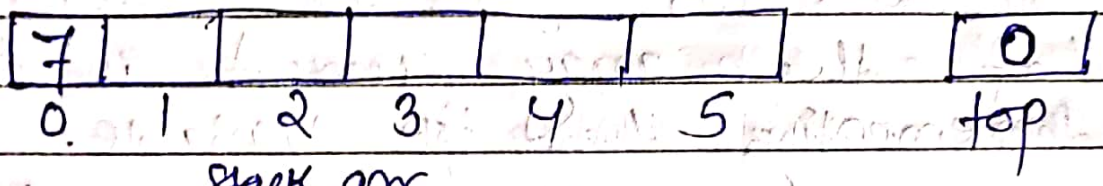


fig:- Empty stack. #define MAX 6

To indicate that the stack is empty, we will put -1 in the variable top.

For the push operation:

→ top is incremented by 1.
→ New element is pushed at the position top.



push(7);

After, top[5] pushing the element is not possible that condition is called stack overflow.

For the pop operation:

→ The element at the position of top is deleted.
→ top is decremented by 1.

After, top[-1] poping the element is not possible that condition is called stack underflow.

Q. stack_arr

1	2	3	4
0	1	2	3

3
top

How to delete the element at index 3?

→ We can't simply remove the array element. Still, we can give the user an illusion that the array element is deleted by decrementing the top variable.

top variable always keeps track of the topmost element of the stack. If the top is decremented by 1, then user will perceive that the topmost element is deleted. Now, we will write a pop function which is capable of deleting the topmost element of the stack and returns the deleted element.

Q. If the sequence of operation - push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2), pop are performed on a stack, the sequence of popped out values.

A) 2, 2, 1, 1, 2
B) 2, 2, 1, 2, 2
C) 2, 1, 2, 2, 1
D) 2, 1, 2, 2, 2

Ans 1 x x 2, 1 x, 2, 2 x x
⇒ 2, 2, 1, 1, 2.

Q. Consider the following operations performed on a stack of size 5: push(a); pop(); push(b); push(); pop(); push(d); pop(); pop(); push(e) which of the following statement is correct?

A) underflow occurs
B) ~~Stack~~ operations are performed smoothly
C) overflow occurs
D) none of the mentioned

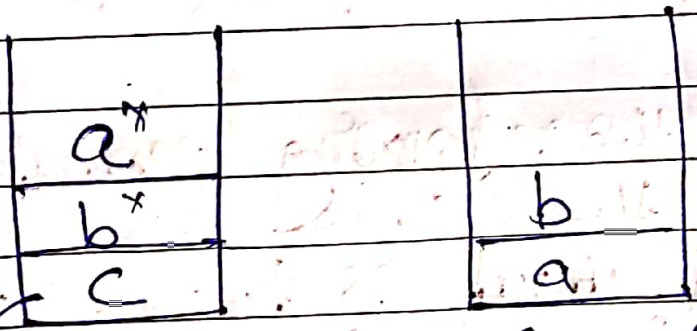
a, b, c, d, e

Q. which of the following permutation can be obtained in the same order using a stack assuming that input is the sequence 5,6,7,8,9 in that order?

A) 7,8,9,5,6
B) 5,9,6,7,8
C) 7,8,9,6,5
D) 9,8,7,5,6

Q. Stack A has the entries a, b, c (with a on top). Stack B is empty. An entry popped out of stack A can be pushed immediately or pushed to stack B. An entry popped out of the stack B can be printed. In this arrangement, which of the following permutations of a, b, c are not possible?

A) b, a, c
B) b, c, a
C) c, a, b
D) a, b, c



o/p :- c, b, a which is not possible.