

1. Algorithm for POP

Suppose $STACK[SIZE]$ is a one dimensional array for implementing stack.

Step 1. If $TOP < 0$ then:

a. display stack is empty
b. exit.

Step 2. Else remove the top most element.

Step 3. $data = STACK[TOP]$

Step 4. $TOP = TOP - 1$

Step 5. Exit.

2. Algorithm for Push.

Suppose $STACK[SIZE]$ is a one dimensional array for implementing stack.

Step 1. If $TOP = SIZE - 1$ then:

a. Display "The stack is in overflow condition"
b. exit

Step 2. ~~Else~~ $TOP = TOP + 1$

Step 3. $STACK[TOP] = Item$ (given)

Step 4. Exit.

3.6 Algorithm to display elements of a stack.

Step-1 Check whether stack is empty. ($top == -1$)

Step-2 If it is empty, display "stack is empty."
and terminate the function.
Exit.

Step-3 If not empty, then define a variable 'i' & initialise with top. Display STACK[i] value and decrement 'i' value by one. ($--i$).

Step-4 Repeat Step-3 until i becomes 0.