Stack Definition

A stack, linear data structure, is a collection of similar type of elements called nodes. Each node in the stack is divided into two parts, where the first part stores the information and the other part stores the address of next node in the stack. There is a limitation about the operations that can be performed on stack. The elements of stack can only be accessed (inserted/ removed) from one side only. This side is sometimes referred as 'top'. The insertion of an element at stack top is referred as '**push()**' and the removal of element from stack top is referred as '**pop()**'. Since the element inserted in the **last** remains at the top of the stack it will be the **first** to be removed. Hence, stack is also known as LIFO data structure. The other operations that can be performed on the stack are: '**peek()**' that displays only the topmost element in the stack.

Stack can be represented in two ways:

    i)   By using an array
    ii)   By using a linked list

When stack is implemented by using an array, there will be some limitations enforced on such stack that its size cannot be changed during the run-time by executing the operations on it, from the size that has been declared. The certain limitations that have to be kept in mind while implementing a stack using an array are as:

    i)   The size of the stack cannot be changed dynamically as it remains static as specified during compile time according to the size of the array.
    ii)   To represent empty stack top position of stack should be set as -1.
    iii)   The stack should be checked to be overflowed while pushing an element.
    iv)   The stack should be checked for underflow while popping and peeking any element.

Push operation on array stack

Algorithm

begin

if <Stack_top>=SIZE then PROMPT "Full stack"

<Stack_top>:=<Stack_top>+1;

Stack(<Stack_top>):=information;

end

Pop operation on array stack

Algorithm

begin

if <Stack_top>=-1 then PROMPT "Empty Stack"

information:=Stack(<Stack_top>);

<Stack_top>:=<Stack_top>-1;

end

Peek operation on array stack

Algorithm

begin

if <Stack_top>=-1 then PROMPT "Empty Stack"

information:=Stack(<Stack_top>);

PROMPT information;

end