Algorithm_Stack_Push_Using_Array()

Input: A Stack implemented using an array, say A, an element to be pushed, say ITEM and TOP, a variable that will hold the index of the item pushed last in the stack.

Output: ITEM successfully pushed at the TOP^{th} position of the stack otherwise suitable overflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

1. Begin
2. If (TOP = U)
3. Then
4. Print “Stack overflow, ITEM can’t be pushed in stack”
5. Else
6. Set TOP = TOP + 1
7. Set A[TOP] = ITEM
8. End If
9. End

Note: The initial value of TOP will be L – 1 when the stack is empty

Algorithm_Stack_Pop_Using_Array()

Input: A Stack implemented using an array, say A and TOP, a variable that will hold the index of the item pushed last in the stack.

Output: ITEM successfully popped from the TOP^{th} position of the stack otherwise suitable overflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

```
If TOP = L – 1
Then
        Print “Stack underflow, no item to pop”
Else
        Set ITEM = A[TOP]
        Set TOP = TOP – 1
        Return ITEM
End If
End
```

Note: The initial value of TOP will be L – 1 when the stack is empty

Algorithm_Stack_Traverse_Using_Array()

Input: A Stack implemented using an array, say A and TOP, a variable that will hold the index of the item pushed last in the stack.

Output: The elements of the stack are successfully traversed from the TOPth position till the item pushed first in the stack otherwise suitable underflow message.

Data Structure used: An array A[L..U] where L = Lower index of the array, U = Upper index of the array and SIZE = U – L + 1

Steps:

1. If TOP = L – 1
2. Then
3. Print “Stack underflow, no item to traverse”
4. Else
5. Set i = TOP
6. While i ≥ L
7. Begin
8. Process (A[i]) // Process() is a procedure that processes the element being traversed in required way
9. Set i = i – 1
10. End While
11. End if
12. End

Note: The initial value of TOP will be L – 1 when the stack is empty