**THEORY:**  
A stack is a data structure that follows the Last In, First Out (LIFO) principle, meaning that the last element added to the stack is the first one to be removed. Think of it like a stack of plates – you add a plate to the top of the stack, and when you want to remove one, you take it from the top.

**Basic Stack Operations:**

1. Push:

* The push operation adds an element to the top of the stack.
* It increments the stack size by 1.
* The newly added element becomes the top of the stack.

2. Pop:

* The pop operation removes the element from the top of the stack.
* It decrements the stack size by 1.
* The element that is removed is the one that was last added (LIFO).

3. Peek (or Top):

* The peek operation returns the element at the top of the stack without removing it.
* It allows you to examine the top element without modifying the stack.

**PROBLEM ANALYSIS:**

The problem is to create a stack of a definite size defining necessary functions for the operation of a stack and performing the operations as per the user input.

**ALGORITHM:**

Step 1 – Start

Step 2 – Declare and initialize all the necessary variables(top=-1, item, stack, stack\_num, stack[stack\_num]

Step 3 – For push operation: if top == stack\_num -1 display "Stack is full" else top++ and stack[top] = data

Step 4 – For next push operation repeat step 3

Step 5 – For pop operation: if top == -1 display "Stack empty" else display stack[top] and top

Step 6 –For next pop operation repeat step 5

Step 7 –For display operation: if top == -1 display "Stack empty" else for (i = top ; i>=0; i—) display stack[i]

Step 8–For peek operation: if top == -1 display "Stack empty" else display stack[top] Step 9–End

**Question:**

Write a menu driver program to perform all the operations in Stack.

1. PUSH
2. POP
3. DISPLAY
4. NO. OF ELEMENTS
5. PEEK