DATA-STRUCTURES

M.J.N.V.Sai
20BWIA12A0
IT-B

HOME-ASSIGNMENT-1

1) Show the Detail concept of the stack to evaluate the following Positive expression;

Given.

$$632 - 5 * + 1 \wedge 7 + *$$

Algorithm : To evaluate a post-fix expression.

Step 1 : Input is a post-fix expression.

Step 2 : Read the expression from Left to Right.

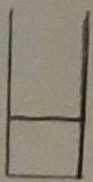
Step 3 : If operand is encountered move to the stack.

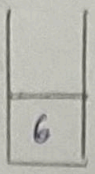
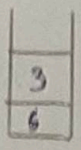
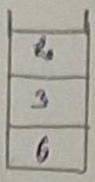
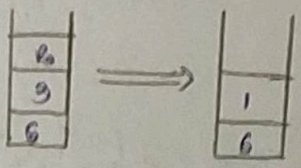
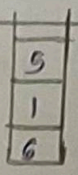
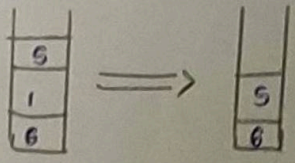
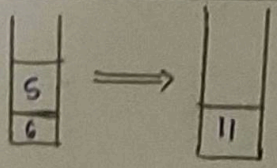
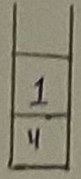
Step 4 : If operator is encountered pop() for 2 Times, & Apply the operator and move the Result to stack.

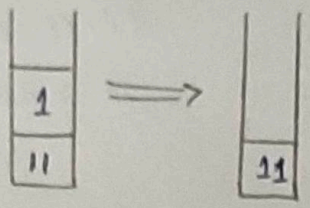
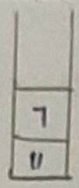
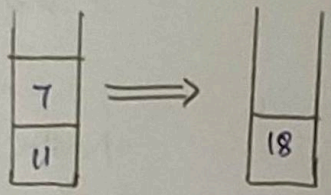
Step 5 : Continue the process, until the end of expression.

Step 6 : output is post-fix value.

Evaluation : $632 - 5 * + 1 \wedge 7 + *$

Read	Stack	computation
-		At very Beginning Stack is Empty.

6		Since 6 is an operand, It moves into a stack
3		Since 3 is an operand, It moves into a stack
2		Since 2 is an operand, It moves into a stack
-		operator is encountered Apply POP() 2 Times V_1 = POP(); V_2 = POP() Push ($V_2 - V_1$) into stack
5		Since 5 is an operand It moves into a stack
*		operator is encountered Apply POP() 2 Times, V_1 = POP(); V_2 = POP() Push ($V_2 * V_1$) into a stack.
+		operator is encountered Apply POP() 2 Times, V_1 = POP(); V_2 = POP(); Push ($V_2 + V_1$) into stack
1		Since 1 is an operand, It moves into a Stack.

^		operator is encountered Apply POP() 2 Times, v_1 = POP(); v_2 = POP(); push(v_2 ^ v_1) into a stack
7		Since 7 is an operand it is moved into a stack.
+		operator is encountered Apply POP() 2 Times, v_1 = POP(); v_2 = POP(); push(v_2 + v_1) into a stack.

RESULT : Post - Evaluation value : 18

$\therefore$ 6 3 2 - 5 * + 1 ^ 7 + = 18