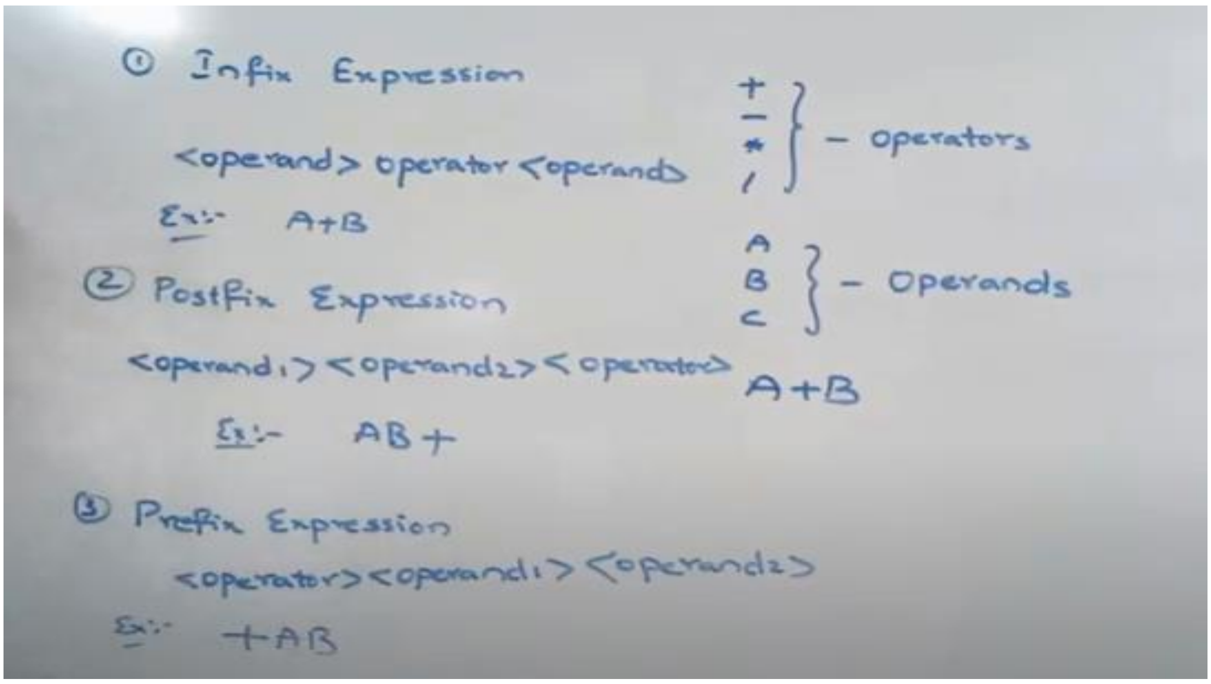

① Infix Expression

<operand> operator <operand>

Ex:- A+B

② Postfix Expression

<operand1> <operand2> <operator>

Ex:- AB+

③ Prefix Expression

<operator> <operand1> <operand2>

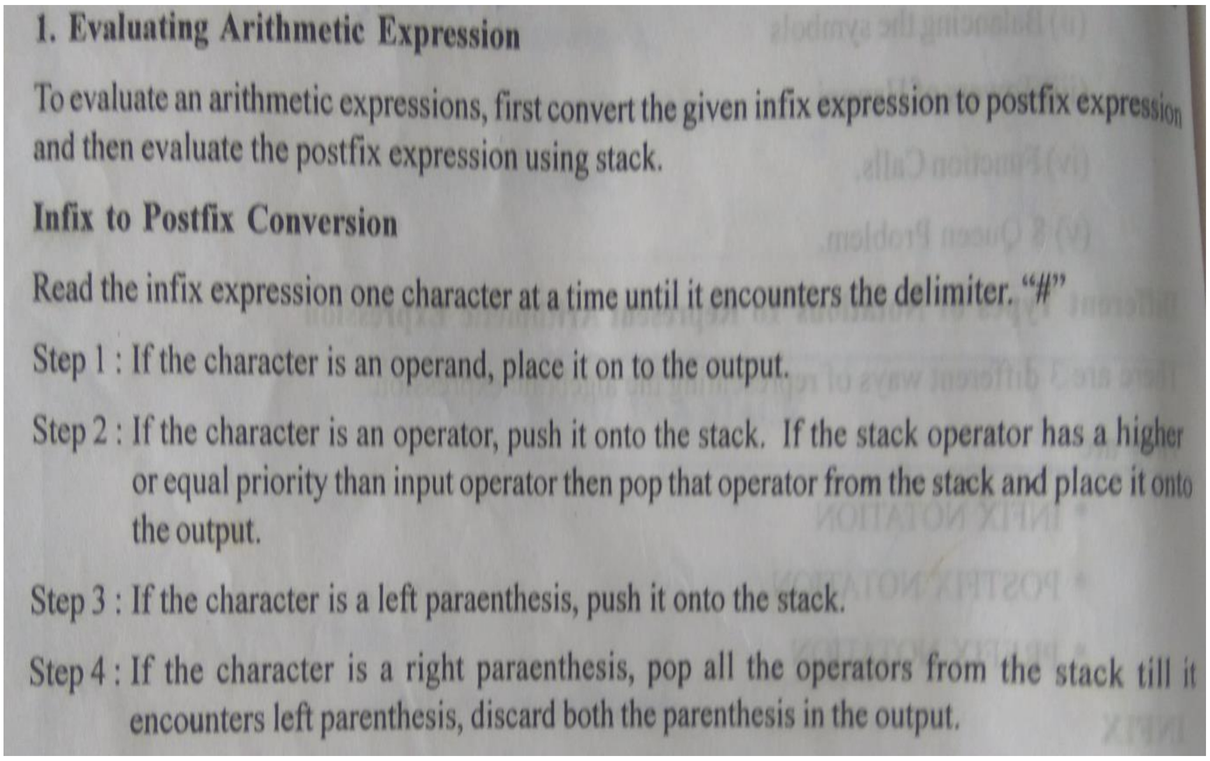
Ex:- +AB

+, −, *, / } − operators

A, B, C } − Operands

A+B

Infix to Postfix Conversion



1. Evaluating Arithmetic Expression

To evaluate an arithmetic expressions, first convert the given infix expression to postfix expression and then evaluate the postfix expression using stack.

Infix to Postfix Conversion

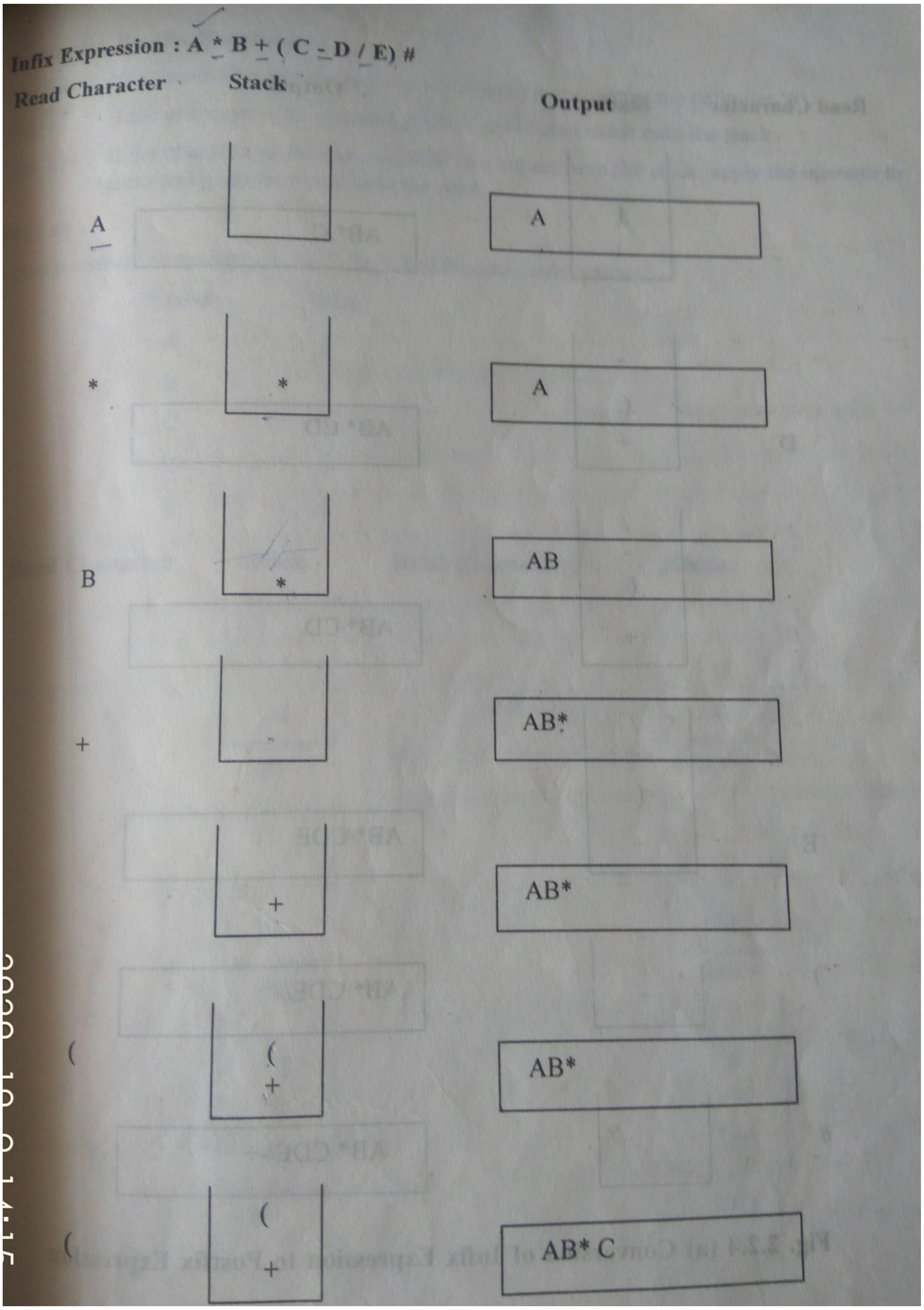
Read the infix expression one character at a time until it encounters the delimiter. "#"

Step 1 : If the character is an operand, place it on to the output.

Step 2 : If the character is an operator, push it onto the stack. If the stack operator has a higher or equal priority than input operator then pop that operator from the stack and place it onto the output.

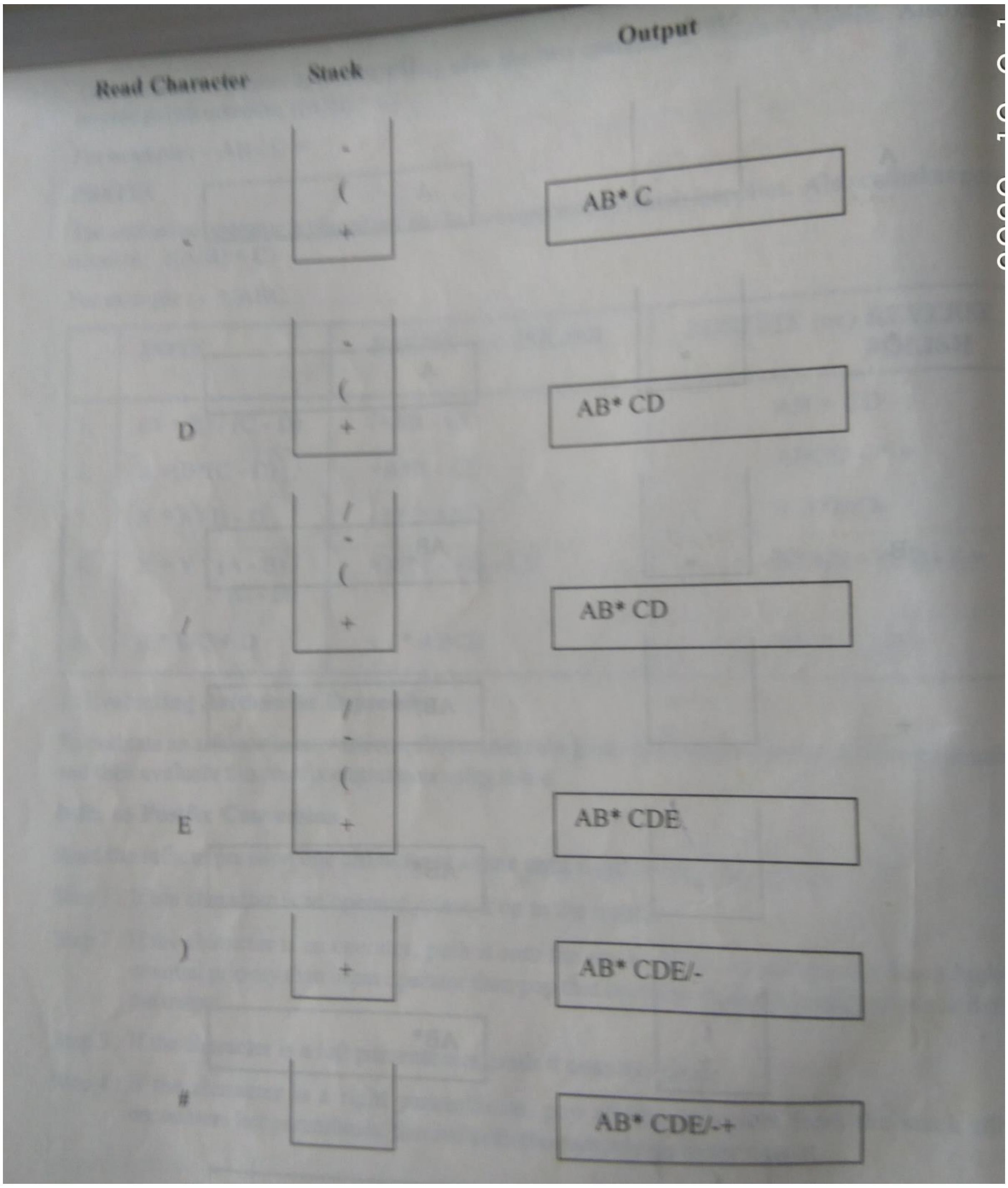
Step 3 : If the character is a left paraenthesis, push it onto the stack.

Step 4 : If the character is a right paraenthesis, pop all the operators from the stack till it encounters left parenthesis, discard both the parenthesis in the output.

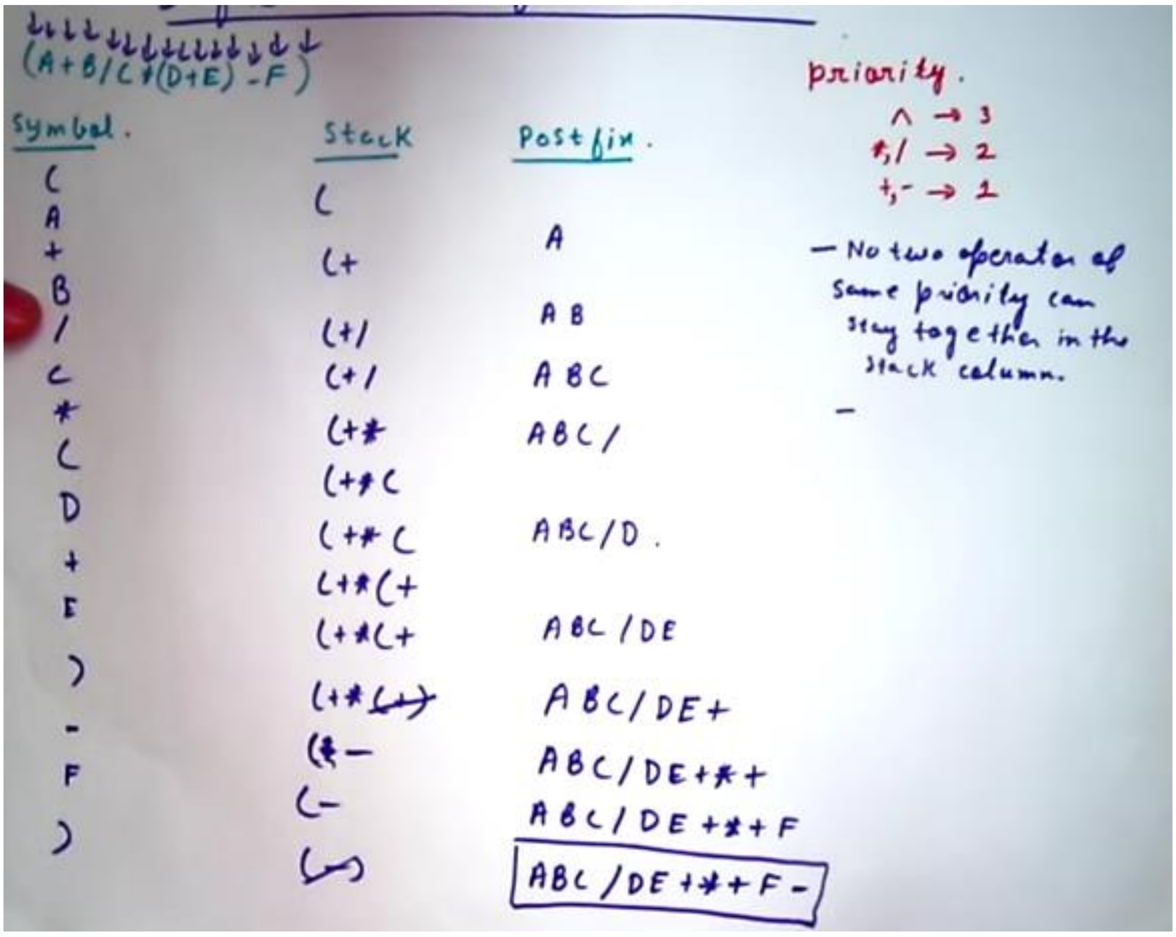


Infix Expression : A * B + (C - D / E) #

Read Character	Stack	Output
A		A
*	*	A
B	*	AB
+		AB*
	+	AB*
(	(+	AB*
(	(+	AB* C



Read Character	Stack	Output
-	- (+	AB* C
D	- (+	AB* CD
/	/ - (+	AB* CD
E	/ - (+	AB* CDE
)	+	AB* CDE/-
#		AB* CDE/-+



(A+B/C*(D+E)-F)

Symbol.	Stack	Postfix.
(	(	
A		A
+	(+	
B		A B
/	(+/	
C	(+/	A B C
*	(+*	A B C /
(	(+*(	
D	(+*(	A B C / D .
+	(+*(+	
E	(+*(+	A B C / D E
)	(+*~~(+)~~	A B C / D E +
-	(*-	A B C / D E + * +
F	(-	A B C / D E + * + F
)	(-)	$\boxed{ABC/DE+*+F-}$

priority.

^ → 3

*,/ → 2

+,- → 1

— No two operator of same priority can stay together in the stack column.

—

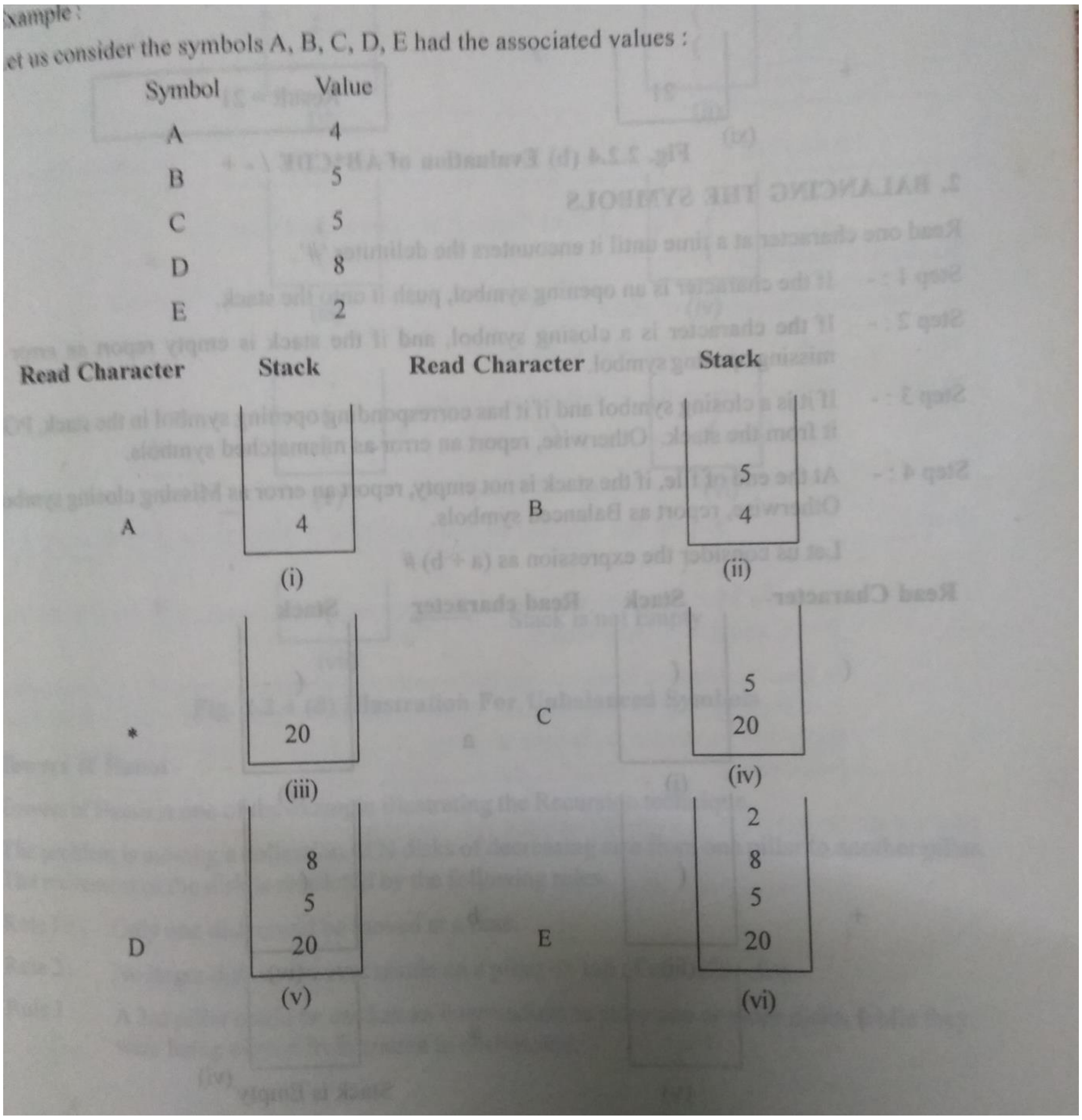
Evaluating Postfix Expression

Read the postfix Expression one character at a time until encounters the delimiter “ #”

Step 1:If the character is an operand , push its associated value onto the stack

Step 2: If the character is an operator, pop two values from the stack , apply the operator to them and push the result onto the stack

AB*CDE/-+



xample :

et us consider the symbols A, B, C, D, E had the associated values :

Symbol	Value
A	4
B	5
C	5
D	8
E	2

Read Character
Stack
Read Character
Stack
A
4
(i)
B
5
4
(ii)
*
20
(iii)
C
5
20
(iv)
D
8
5
20
(v)
E
2
8
5
20
(vi)

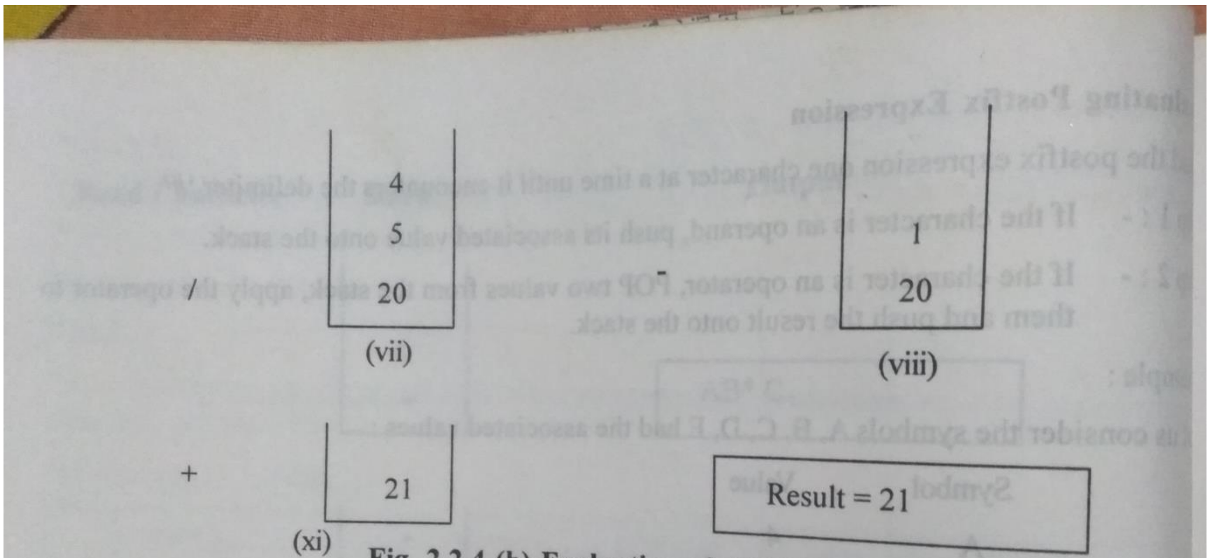


Fig. 2.2.4 (b) Evaluation of AB*CDE / - +

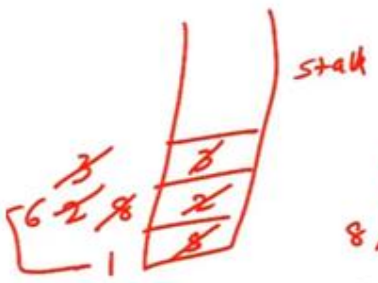
The following postfix expression with single digit operands is evaluated using a stack:

8 2 3 ^/ 2 3*+5 1*-

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are

a) 6,1 b) 5,7

c) 3,2 d) 1,5



$2 \wedge 3 = 8$

$8 / 8 = 1$

$2 * 3 = 6$