Different Types of Notations To Represent Arithmetic Expression

There are 3 different ways of representing the algebraic expression.

They are

* INFIX NOTATION
* POSTFIX NOTATION
* PREFIX NOTATION

INFIX

In Infix notation, The arithmetic operator appears between the two operands to which it is being applied.

For example : - A / B + C

POSTFIX

The arithmetic operator appears directly after the two operands to which it applies. Al reverse polish notation. ((A/B) + C)

For example : - AB / C +

PREFIX

The arithmetic operator is placed before the two operands to which it applies. Also called notation. ((A/B) + C)

For example : - +/ABC

	INFIX	PREFIX (or) POLISH	POSTFIX (or) REVER POLIS
1.	(A + B) / (C - D)	/+AB - CD	AB + CD - /
2.	A + (B*(C - D))	+A*B - CD	ABCD - * +
3.	(X * A) / B - D)	- / * XABD	X A*B/D-
4.	X + Y * (A - B) / (C - D)	+X/*Y - AB - CD	XYAB - *CD - /
5.	A * B/C + D	+ / * ABCD	AB * C / D +

1. Evaluating Arithmetic Expression

To evaluate an arithmetic expressions, first convert the given infix expression to postfix expr 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. "#"

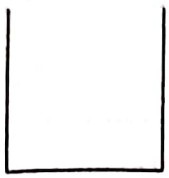
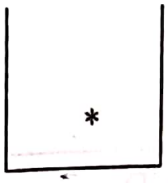
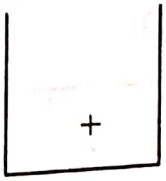
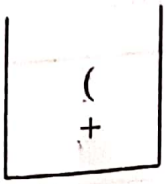
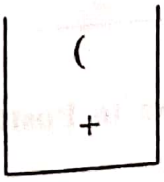
Step 1 : If the character is an operand, place it on to the output.

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

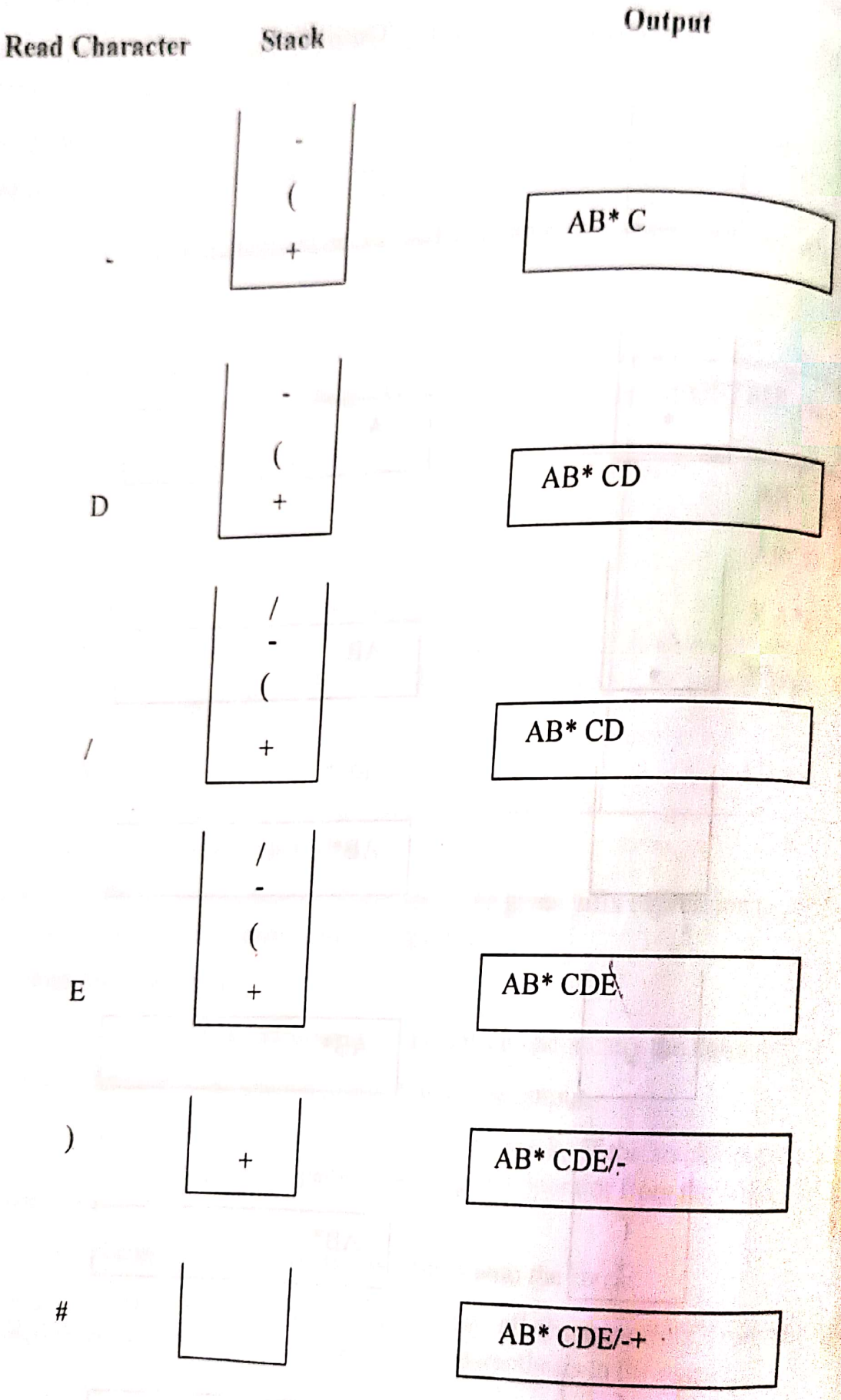


Fig. 2.2.4 (a) Conversion of Infix Expression to Postfix Expression

Evaluating Postfix Expression

Read the postfix expression one character at a time until it 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.

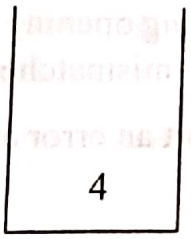
Example :

Let 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

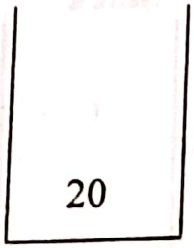
Read Character **Stack**

A



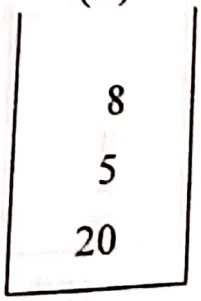
(i)

*



(iii)

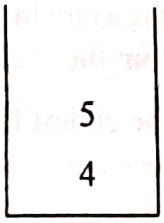
D



(v)

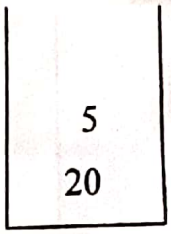
Read Character **Stack**

B



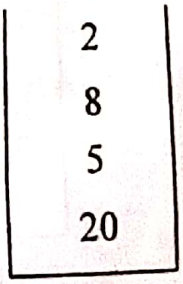
(ii)

C



(iv)

E



(vi)

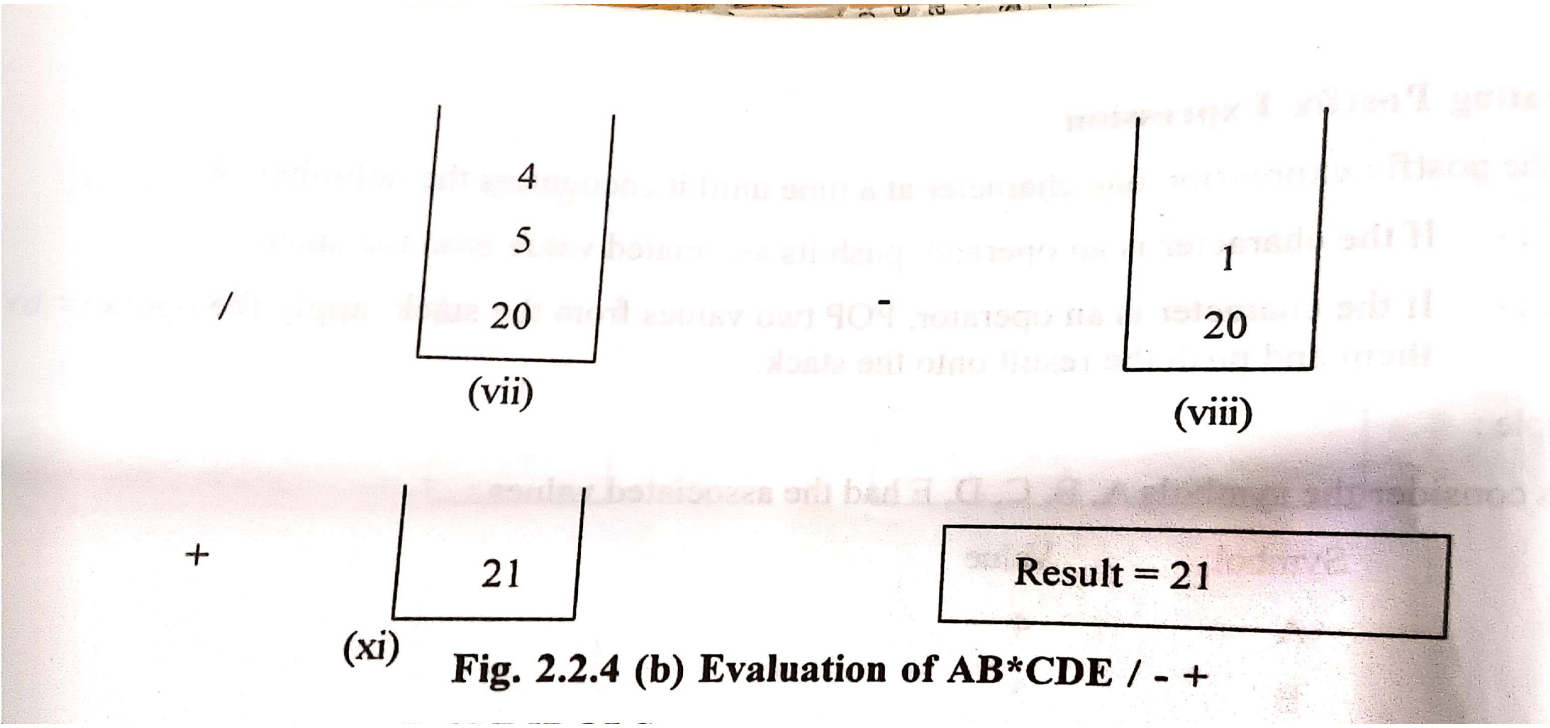


Fig. 2.2.4 (b) Evaluation of AB*CDE / - +