

## Stack

① Algorithm to push an element in a stack

Step 1: IF  $TOP = MAX - 1$ , THEN

PRINT "OVERFLOW"

Go to Step 4

[END OF IF]

Step 2: SET  $TOP = TOP + 1$

Step 3: SET  $STACK[TOP] = VALUE$

Step 4: END

② Algorithm to pop an element from stack

Step 1: IF  $TOP = NULL$ , then

PRINT "UNDERFLOW"

Go to Step 4.

[END OF IF]

Step 2: SET  $VAL = STACK[TOP]$

Step 3: SET  $TOP = TOP - 1$

Step 4: END

③ Algorithm for peek operation.

Step 1: IF  $TOP = NULL$ , then

PRINT "STACK IS EMPTY"

Go to Step 3

[END OF IF]

Step 2: RETURN  $STACK[TOP]$

Step 3: END

# conversion of expression through stack

① Postfix to Infix

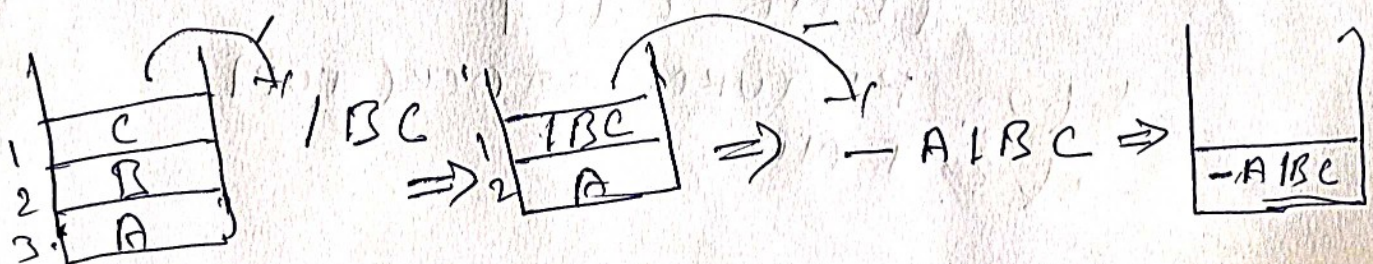
Ex  $AB - DE + F * /$

| Reading Postfix | Stack top           | Expression            |
|-----------------|---------------------|-----------------------|
| A               | A                   | [A]                   |
| B               | B                   | [B]<br>[A]            |
| -               | A-B                 | [A-B]                 |
| D               | D                   | [D]<br>[A-B]          |
| E               | E                   | [E]<br>[D]<br>[A-B]   |
| +               | D+E                 | [D+E]<br>[A-B]        |
| F               | F                   | [F]<br>[D+E]<br>[A-B] |
| *               | (D+E) * F           | [(D+E) * F]<br>[A-B]  |
| /               | (A+B) / ((D+E) * F) | [(A+B) / ((D+E) * F)] |
|                 |                     | infix                 |

② Postfix to prefix

operators + operands + operand,

Ex  $ABC1-AK/L-*$

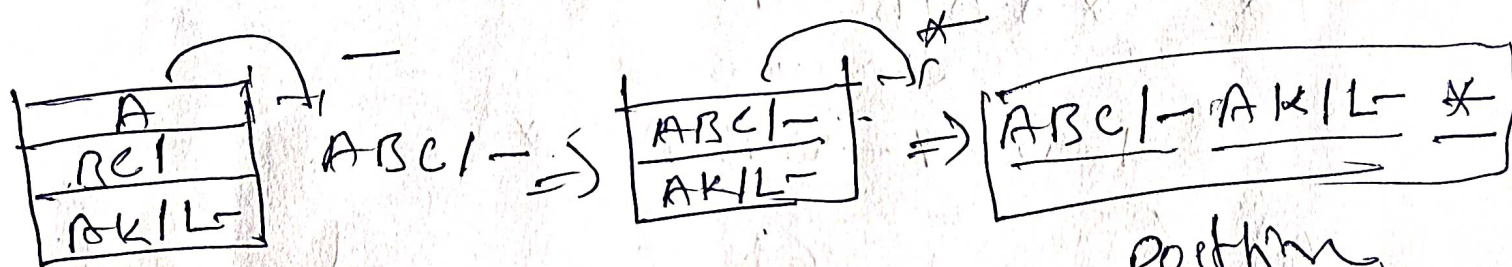
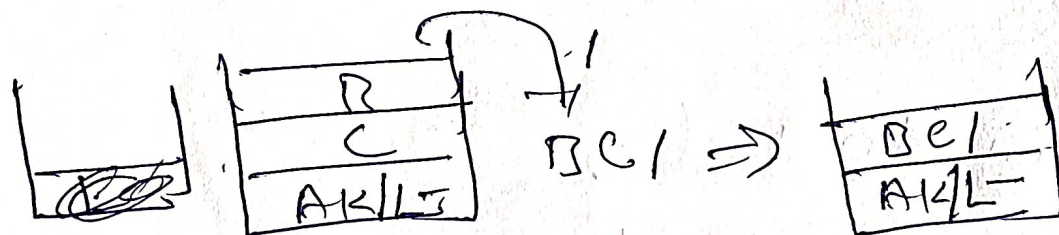
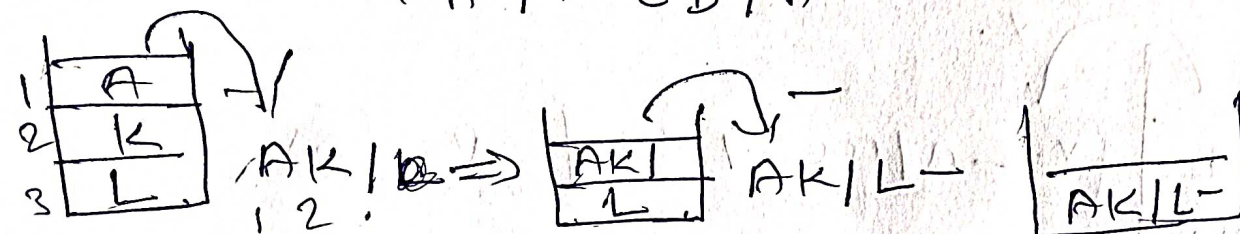


a) Prefix to postfix

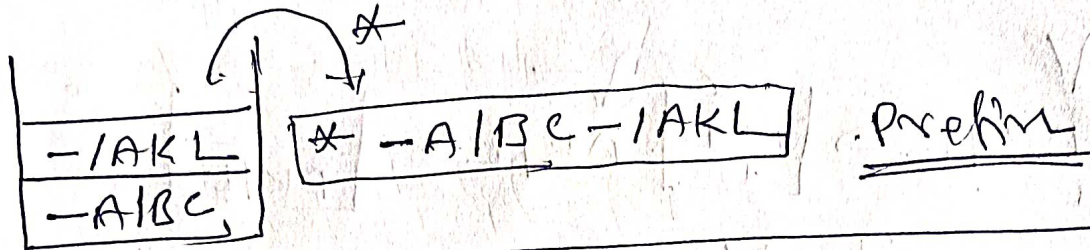
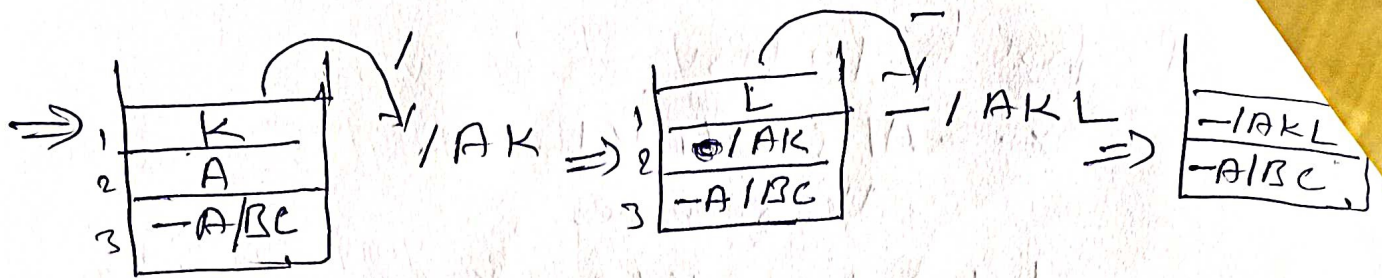
\* - A B C - / A K L ( ) ( )

⇓ reverse

L K A / - C B / A - \*



Postfix

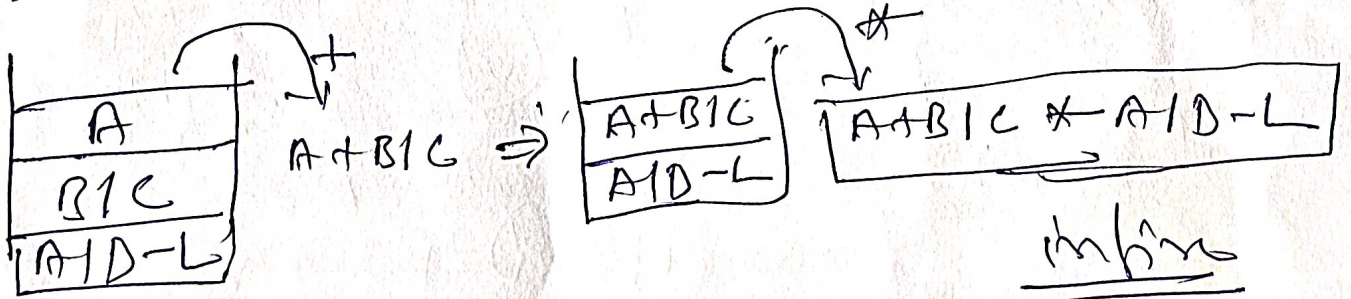
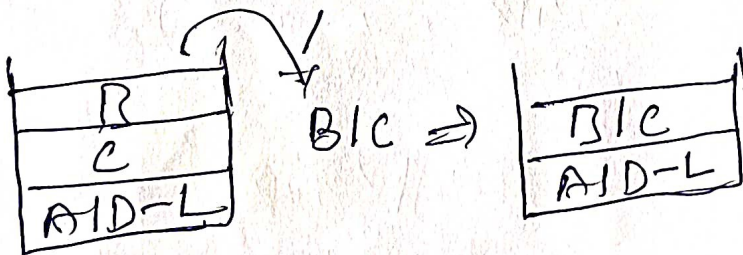
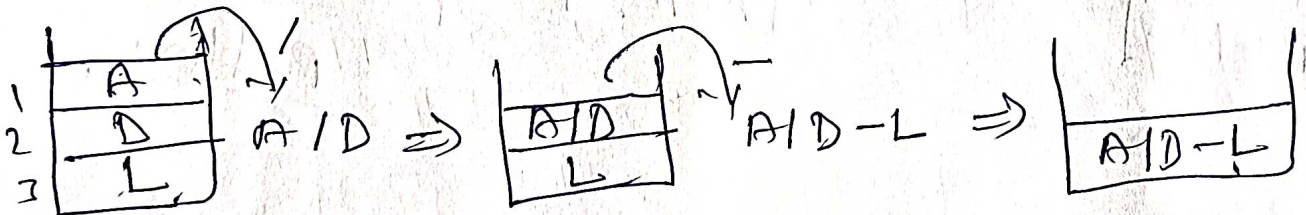


### (c) Prefix to Infix

Ex  $* + A/BC - /ADL$

↓ reverse

$LDA / - C B / A + *$



convert infix to postfix expression.

Ex  $(9 - ((3 * 4) + 8) / 4)$  infix

| infix | stack    | postfix           |
|-------|----------|-------------------|
| (     | (        |                   |
| 9     | (        | 9                 |
| -     | ( -      | 9                 |
| (     | ( - (    | 9                 |
| (     | ( - ((   | 9                 |
| 3     | ( - ((   | 9 3               |
| *     | ( - (( * | 9 3 *             |
| 4     | ( - (( 4 | 9 3 4             |
| )     | ( - (    | 9 3 4 *           |
| +     | ( - ( +  | 9 3 4 *           |
| 8     | ( - ( +  | 9 3 4 * 8         |
| )     | ( -      | 9 3 4 * 8 +       |
| /     | ( - /    | 9 3 4 * 8 +       |
| 4     | ( - /    | 9 3 4 * 8 + 4     |
| )     |          | 9 3 4 * 8 + 4 / - |

Postfix

## Evaluation of postfix expression —

9 3 4 \* 8 + 4 / -

| <u>Symbol</u> | <u>Stack</u> |
|---------------|--------------|
| 9             | 9            |
| 3             | 9 3          |
| 4             | 9 3 4        |
| *             | 9 12         |
| 8             | 9 12 8       |
| +             | 9 20         |
| 4             | 9 20 4       |
| /             | 9 5          |
| -             | 4            |