

CONVERT THE EXPRESSION FROM INFIX TO POSTFIX: -

```
1  #include <stdio.h>
2  #include <ctype.h>
3  #include <string.h>
4
5  #define SIZE 50
6
7  char stack[SIZE];
8  int top = -1;
9
10 void push(char elem) {
11     if (top >= SIZE - 1) {
12         printf("Stack overflow\n");
13         return;
14     }
15     stack[++top] = elem;
16 }
17
18 char pop() {
19     if (top == -1) {
20         printf("Stack underflow\n");
21         return '#';
22     }
23     return stack[top--];
24 }
25
26 int pr(char symbol) {
27     if (symbol == '^') {
28         return 3;
29     } else if (symbol == '*' || symbol == '/') {
30         return 2;
31     } else if (symbol == '+' || symbol == '-') {
32         return 1;
33     } else {
34         return 0;
35     }
36 }
```

```

37
38 int main() {
39     char infix[SIZE], postfix[SIZE];
40     char ch, elem;
41     int i = 0, k = 0;
42
43     printf("Enter infix expression: ");
44     if (fgets(infix, sizeof(infix), stdin) == NULL) {
45         printf("Error reading input\n");
46         return 1;
47     }
48     infix[strcspn(infix, "\n")] = '\0';
49
50     push('#');
51
52     while ((ch = infix[i++]) != '\0') {
53         if (ch == '(') {
54             push(ch);
55         } else if (isalnum(ch)) {
56             postfix[k++] = ch;
57         } else if (ch == ')') {
58             while (stack[top] != '(') {
59                 postfix[k++] = pop();
60             }
61             elem = pop();
62         } else {
63             while (pr(stack[top]) > pr(ch) || (pr(stack[top]) == pr(ch) && ch != '^')) {
64                 postfix[k++] = pop();
65             }
66             push(ch);
67         }
68     }
69
70     while (stack[top] != '#') {
71         postfix[k++] = pop();
72     }
73     postfix[k] = '\0';
74
75     printf("\nPostfix Expression = %s\n", postfix);
76
77     return 0;
78 }
79

```

OUTPUT: -

```
Enter infix expression: A+B*C-D
```

```
Postfix Expression = ABC*+D-
```

```
Process returned 0 (0x0)   execution time : 7.112 s
Press any key to continue.
```