

11-10-2025

2) WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide)

```
#include <stdio.h>
#include <ctype.h>
#define MAX 100
char stack[MAX];
int top = -1;
/* Function to push into stack */
void push(char ch) {
    stack[++top] = ch;
}
/* Function to pop from stack */
char pop() {
    return stack[top--];
}
/* Function to check operator precedence */
int precedence(char ch) {
    if (ch == '+' || ch == '-')
        return 1;
    if (ch == '*' || ch == '/')
        return 2;
    return 0;
}
int main() {
    char infix[MAX], postfix[MAX];
    int i = 0, j = 0;
    char ch;
    printf("Enter infix expression: ");
    scanf("%s", infix);
}
```

```

while (infix[i] != '\0') {
    ch = infix[i];
    /* If operand, add to postfix */
    if (isalnum(ch)) {
        postfix[j++] = ch;
    }
    /* If '(', push to stack */
    else if (ch == '(') {
        push(ch);
    }
    /* If ')', pop until '(' */
    else if (ch == ')') {
        while (stack[top] != '(')
            postfix[j++] = pop();
        pop(); // remove '('
    }
    /* If operator */
    else {
        while (top != -1 && precedence(stack[top]) >= precedence(ch))
            postfix[j++] = pop();
        push(ch);
    }
    i++;
}
/* Pop remaining operators */
while (top != -1) {
    postfix[j++] = pop();
}
postfix[j] = '\0';
printf("Postfix expression: %s\n", postfix);
return 0;
}

```

```
C:\Users\BMSCLC16\OneDrive - + ~  
Enter infix expression: a+b(c-d)*f  
Postfix expression: abcd-f++  
Process returned 0 (0x0) execution time : 21.478 s  
Press any key to continue.
```

