```
/* DS Lab 4
Develop a Program in C for converting an Infix
Expression to Postfix Expression. Program
should support for both parenthesized and free
parenthesized
expressions with the operators:+, -,*,/,%
(Remainder), A (Power) and alphanumeric
operands
*/

#include <stdio.h>
#include <stdlib.h>

int F(char symbol)
{
    switch(symbol)
    {
        case '#': return -1;

        case '+':
        case '-': return 2;

        case '*':
        case '/': return 4;

        case '$':
        case '^': return 5;

        default: return 8;

        case '(': return 0;
    }
}

int G(char symbol)
{
    switch(symbol)
    {
```

```
        case ')': return 0;

        case '+':
        case '-': return 1;

        case '*':
        case '/': return 3;

        case '$':
        case '^': return 6;

        default: return 7;

        case '(': return 9;
    }
}

void infix_2_postfix(char infix[], char postfix[])
{
    int top, i, j = 0;
    char s[20];

    top = -1;
    s[++top] = '#';

    for(i = 0; infix[i] != '\0'; i++)
    {
        while(F(s[top]) > G(infix[i]))
            postfix[j++] = s[top--];

        if(F(s[top]) != G(infix[i]))
            s[++top] = infix[i];
        else
            top--;
    }

    while(s[top] != '#')
        postfix[j++] = s[top--];
```

```
    postfix[j] = '\0';
}

void main()
{
    char infix[20], postfix[20];

    printf("Enter Infix Expression: ");
    scanf("%s", infix);

    infix_2_postfix(infix, postfix);

    printf("\nInfix to Postfix Converted Expression
    : %s\n", postfix);
}

/*
Output:

Enter Infix Expression : a+b*(c/d$e$f)*(a-b)/c

Infix to Postfix Converted Expression :
abcdef$$/*ab-*c/+

Enter Infix Expression : X^Y^Z-M+N+P/Q

Infix to Postfix Converted Expression :
XYZ^^M-N+PQ/+

*/
```