

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <process.h>
```

```
int F (char symbol)
```

```
// stack precedence : F
```

```
{
```

```
switch (symbol)
```

```
{
```

```
case '+' :
```

```
case '-' : return 2;
```

```
case '*' :
```

```
case '/' : return 4;
```

```
case '^' :
```

```
case '$' : return 5;
```

```
case '[' : return 6;
```

```
case '#' : return -1;
```

```
default : return 8;
```

```
}
```

```
}
```

```
int G (char symbol)
```

```
// input string : G
```

```
{
```

```
switch (symbol)
```

```
{
```

```
case '+' :
```

```
case '-' : return 1;
```

```
case '*' :
```

```
case '/' : return 3;
```

```

case '^':
case '$': return 6;
case '(': return 9;
case ')': return 0;
case '#':
default: return 7;

```

```

}
{
}

```

```

void infix_postfix(char infix[], char postfix[])
{

```

```

    int top, i, j;
    char s[30], symbol;
    top = -1;
    s[++top] = '#';
    j = 0;
    for (i = 0; i < strlen(infix); i++)
    {
        symbol = infix[i];
        while (F(s[top]) > G(symbol))
        {
            postfix[j] = s[top--];
            j++;
        }
        if (F(s[top]) != G(symbol))
            s[++top] = symbol;
        else
            top--;
    }

```

```

    while (s[top] != '#')
    {

```

```

        {

```

```

            &postfix[j++] = s[top--];

```

```

        }

```

```

        postfix[j] = '\0';

```

```

    }

```



```
void main()
```

```
{
```

```
    char infix[20];
```

```
    char postfix[20];
```

```
    clrscr();
```

```
    printf("\nEnter the valid infix expression: ");
```

```
    scanf("%s", infix);
```

```
    infix = postfix (infix, postfix);
```

```
    printf("\nThe postfix exp is: ");
```

```
    printf("%s", postfix);
```

```
    getch();
```

```
}
```

```
main.c
52
53     if(F(s[top]) != G(symbol))
54     {
55         s[++top] = symbol;
56     }
57
58     else
59     {
60         top--;
61     }
62 }
63
64 while(s[top] != '#')
65 {
66     postfix[j++] = s[top--];
67 }
68
69 postfix[j] = '\0';
70 }
71
72 void main()
73 {
74     char infix[20];
75     char postfix[20];
76     printf("enter a valid infix expression \n");
77     scanf("%s", infix);
78     infix_postfix(infix, postfix);
79     printf(" the postfix expression is \n");
80     printf("%s", postfix);
81 }
```

```
main.c
29     case '$': return 6;
30     case '(': return 9;
31     case ')': return 0;
32     default : return 7;
33 }
34 }
35
36 void infix_postfix (char infix[], char postfix[])
37 {
38     int top, i , j;
39     char s[30], symbol;
40     top = -1;
41     s[++top] = '#';
42     j = 0;
43     for(i=0; i<strlen(infix); i++)
44     {
45         symbol = infix[i];
46
47         while (F(s[top]) > G(symbol))
48         {
49             postfix[j] = s[top--];
50             j++;
51         }
52
53         if(F(s[top]) != G(symbol))
54         {
55             s[++top] = symbol;
56         }
57     }
```

onlinegdb.com/online_c_compiler



Run Debug Stop Share Save Beautify

Language C



input

enter a valid infix expression

a^b*c-d+e/f(g+h)

the postfix expression is

ab^c*d-egh+f/+

...Program finished with exit code 0

Press ENTER to exit console.

