

## Infix\_to\_Postfix\infix\_to\_postfix.c

```
1 #include <stdio.h>
2 #define Maxsize 10
3 int top = -1;
4 char stack[50];
5
6 // Defining push function
7 void push(char ch)
8 {
9     if (top >= Maxsize - 1)
10    {
11        printf("Stack is full \n");
12    }
13    else
14    {
15        top++;
16        stack[top] = ch;
17    }
18 }
19
20 //Defining pop function
21 char pop()
22 {
23     if (top < 0)
24    {
25        printf("stack is empty.");
26        return 'o';
27    }
28    else
29    {
30        char ch = stack[top];
31        top--;
32        return ch;
33    }
34 }
35
36 //Assigning Priority to each character
37 int priority(char x)
38 {
39     switch (x)
40    {
41     case '^':
42         return 5;
43         break;
44     case '/':
45     case '*':
46         return 4;
47         break;
48     case '+':
```

```
49     case '-':
50         return 3;
51         break;
52     case '(':
53         return 0;
54         break;
55     default:
56         return -1;
57         break;
58     }
59 }
60
61 //Defining infix to postfix function
62 void infixtopostfix(char e[])
63 {
64     char ch;
65     for (int i = 0; e[i] != '\0'; i++)
66     {
67         if ((e[i] >= 'A' && e[i] <= 'Z') || (e[i] >= 'a' && e[i] <= 'z'))
68         {
69             printf("%c", e[i]);
70         }
71         else if (e[i] == '(')
72         {
73             push(e[i]);
74         }
75         else if (e[i] == ')')
76         {
77             ch = pop();
78             while (ch != '(')
79             {
80                 printf("%c", ch);
81                 ch = pop();
82             }
83         }
84         else
85         {
86
87             while (priority(stack[top]) >= priority(e[i]))
88             {
89
90                 printf("%c", ch);
91                 ch = pop();
92             }
93             push(e[i]);
94         }
95     }
96     while (top >= 0)
97     {
98         ch = pop();
```

```
99         printf("%c", ch);
100    }
101 }
102
103
104
105 int main()
106 {
107     char e[Maxsize];
108     printf("Enter a expression : ");
109     scanf("%s", e);
110     infixtopostfix(e);
111     return 0;
112 }
113
```