

```
#include <stdio.h>
#include <stdlib.h>
#include <process.h>
int F (char symbol)
```

```
{
    switch (symbol)
```

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

```
    case '*':
```

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

```
    case '^': return 5;
```

```
    case '$': return 0;
```

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

```
    default : return 8;
```

```
}
```

```
}
```

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

```
{
```

```
    switch (symbol)
```

```
{
```

```
    case '+':
```

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

```
    case '*':
```

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

Case 'x':

Case 's': return 6;

Case '(': return 9;

Case ')': return 0;

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++)

{

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

j++;

if (E(s[top]) != 4 (symbol))

s[++top] = symbol;

else top--;

}


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

```
{
```

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

```
}
```

```
void main()
```

```
{
```

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

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

```
    clrscr();
```

```
    printf("Enter the valid infix expression\n");
```

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

```
    infix_to_postfix(infix, postfix);
```

```
    printf("The postfix expression is\n");
```

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

```
    getch()
```

```
}
```

```
}
```



main.c x

```
1  #include<stdio.h>
2  #include<string.h>
3  int F(char symbol)
4  {
5      switch(symbol)
6      {
7          case '+':
8          case '-':return 2;
9          case '*':
10         case '/':return 4;
11         case '^':
12         case '$':return 5;
13         case '(':return 0;
14         case '#': return -1;
15         default : return 8;
16     }
17 }
18 int G(char symbol)
19 {
20     switch(symbol)
21     {
22         case '+':
23         case '-':return 1;
24         case '*':
```

```

24     case '*':
25     case '/':return 3;
26     case '^':
27     case '&':return 6;
28     case '(':return 9;
29     case ')':return 0;
30     default:return 7;
31     }
32 }
33 void infix_postfix(char infix[],char postfix[])
34 {
35     int top,j,i;
36     char s[30],symbol;
37     top=-1;
38     s[++top]='#';
39     j=0;
40     for(i=0;i<strlen(infix);i++)
41     {
42         symbol=infix[i];
43         while(F(s[top])>G(symbol))
44         {
45             postfix[j]=s[top--];
46             j++;
47         }

```

```

46         j++;
47     }
48     if (F(s[top]) != G(symbol))
49         s[++top] = symbol;
50     else
51         top--;
52 }
53 while (s[top] != '#')
54 {
55     postfix[j++] = s[top--];
56 }
57 postfix[j] = '\0';
58 }
59 void main()
60 {
61     char infix[20];
62     char postfix[20];
63     printf("Enter the valid infix expression\n");
64     scanf("%s", infix);
65     infix_postfix(infix, postfix);
66     printf("The postfix expression is\n");
67     printf("%s\n", postfix);
68 }
69

```

Enter the valid infix expression

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

The postfix expression is

$ab^c * d - ef / gh + / +$

Process returned 0 (0x0) execution time : 34.247 s

Press any key to continue.

```
"C:\Users\karthik sai\Documents\infixpostfix\bin\Debug\infixpostfix.exe"
Enter the valid infix expression
X^Y^Z-M+N+P/Q
The postfix expression is
XYZ^^M-N+PQ/+
Process returned 0 (0x0)   execution time : 26.585 s
Press any key to continue.
```