



```
1 #include<stdio.h>
2 #include<string.h>
3
4 int F(char symbol)
5 {
6     switch(symbol)
7     {
8
9         case '+':
10        case '-': return 2;
11        case '*':
12        case '/':return 4;
13        case '^':
14        case '$':return 5;
15        case '(':return 0;
16        case '#':return -1;
17        default:return 8;
18    }
19 }
20 int G(char symbol)
21 {
22     switch(symbol)
23     {
24
25        case '+':
26        case '-': return 1;
27        case '*':
28        case '/':return 3;
29        case '^':
30        case '$':return 6;
31        case '(':return 9;
32        case ')':return 0;
33        default:return 7;
34    }
35 }
36 void infix_postfix(char infix[],char postfix[])
```

```

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

```

```

42 j=0;
43 for(i=0;i<strlen(infix);i++)
44 {
45     symbol = infix[i];
46     while(F(s[top])>G(symbol))
47     {
48         postfix[j]=s[top--];
49         j++;
50     }
51     if(F(s[top])!=G(symbol))
52         s[++top]=symbol;
53     else
54         top--;
55 }
56
57 while(s[top]!='#')
58 {
59     postfix[j++]=s[top--];
60 }
61 postfix[j]='\0';
62 }
63
64 int main()
65 {
66     char infix[20];
67     char postfix[20];
68     printf("Enter the valid infix expression\n");
69     scanf("%s",infix);
70     infix_postfix(infix,postfix);
71     printf("The postfix expression is:\n");
72     printf("%s\n",postfix);
73 }
74
75 }
76
77

```

mpiler

RunDebugStopShareSaveBeautify

Language C

```
main.c
42 j=0;
43 for(i=0;i<strlen(infix);i++)
44 {
45     symbol = infix[i];
46     while(F(s[top])>G(symbol))
47     {
48         postfix[j]=s[top--];
49         j++;
50     }
51     if(F(s[top])!=G(symbol))
52         s[++top]=symbol;
53     else
54         top--;
55 }
56
57 while(s[top]!='#')
58 {
59     postfix[j++]=s[top--];
60 }
61 postfix[j]='\0';
62 }
63
64
65 int main()
66 {
67     char infix[20];
68     char postfix[20];
69     printf("Enter the valid infix expression\n");
70
71     input
72     Enter the valid infix expression
73     x^y^z-m+n+p/q
74     The postfix expression is:
75     xyz^^m-n+pq/+
76
77     ...Program finished with exit code 0
78     Press ENTER to exit console.
```

0123456789

File Explorer Spotify Mail Calendar Google Chrome Firefox Edge

12:29 05-10-2020 ENG

```
Untitled - Notepad
File Edit Format View Help
#include<stdio.h>
#include<string.h>

int F(char symbol)
{
    switch(symbol)
    {
        case '+':
        case '-': return 2;
        case '*':
        case '/':return 4;
        case '^':
        case '$':return 5;
        case '(':return 0;
        case '#':return -1;
        default:return 8;
    }
}
int G(char symbol)
{
    switch(symbol)
    {
        case '+':
        case '-': return 1;
        case '*':
        case '/':return 3;
        case '^':
        case '$':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]='#';
```

```

*Untitled - Notepad
File Edit Format View Help
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';
}

int main()
{
    char infix[20];
    char postfix[20];
    printf("Enter the valid infix expression\n");
    scanf("%s",infix);
    infix_postfix(infix,postfix);
    printf("The postfix expression is:\n");
    printf("%s\n",postfix);
}

```

Ln 39, Col 20 100% Windows (CRLF) UTF-8

Type here to search



12:46 05-10-2020 ENG