

main.c

```
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4
5 int F(char symbol)
6 {
7     switch(symbol)
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
21 int G(char symbol)
22 {
23     switch(symbol)
24     {
25         case '+':
26        case '-':return 1;
27        case '*':
28        case '/':return 3;
29        case '^':
```

main.c

```
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
37 void infix_postfix(char infix[],char postfix[])
38 {
39     int top,i,j;
40     char s[30],symbol;
41     top=-1;
42     s[++top]='#';
43     j=0;
44
45     for(i=0;i<strlen(infix);i++)
46     {
47         symbol=infix[i];
48         while(F(s[top])>G(symbol))
49         {
50             postfix[j]=s[top--];
51             j++;
52         }
53
54         if(F(s[top])!=G(symbol))
55             s[++top]=symbol;
```

main.c

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

```
› clang-7 -pthread -lm -o main main.c
```

```
› ./main
```

```
Enter the valid infix expression (a+b)*(c-d)/(e/f)›
```

```
The postfix expression is
```

```
ab+cd-*ef//
```

```
█
```

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

```
int f (char symbol)
{
```

```
    Switch (Symbol)
{
```

Case '+':

Case '-': return 2;

Case '*':

Case '/': return 4;

Case '^':

Case '\$': return 5;

Case 'c': return 0;

Case '#': return -1;

default: return 8;

}

```
(3) Write a C program to implement a function f()
int f (char symbol)
```

{

```
    Switch (Symbol)
{
```

}

Case '+':

Case '-': return 1;

Case '*':

Case '/': return 3;

Case '^':

Case '\$': return 6;

Case 'c': return 9;

Case ')': return 0;

default: return 7;

3

{

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--;

3

while (S[top] != '#')

{

postfix[j + 1] = S[top--];

3

postfix[j] = '\0';

3

void main()

{

```
char *infix[20];
char postfix[20];
prim t f C "Enter the valid infix expression";
scanf f C ".\s", infix);
infix - postfix (infix, postfix);
printf C "The Postfix expression is \n";
printf C ".\n", postfix );
}.
```

Output:

Enter the Valid infix expression :

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

The post fix expression is

ab+cd-*ef//