

```
15:07 2.9KB/s 4G LTE
new*
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 '$': return 5;
13        case '(': return 0;
14        case '#': return -1;
15        default: return 8;
16    }
17 }
18 int G (char symbol)
19 {
20     switch (symbol)
21     {
22         case '-':
23         case '+':return 1;
24         case '*':
25         case '/':return 3;
26         case '^':
27         case '$': return 6;
28         case '(': return 9;
29         case ')':return 0;
30         case '#': return 0;
31         default: return 7;
32     }
33 }
```

```
15:08 3.5KB/s 4G LTE
new*
34 void infix_postfix(char infix[],char postfix[])
35 {
36     int top,i,j;
37     char s[30],symbol;
38     top=-1;
39     s[++top]='#';
40     j=0;
41     for (i=0;i<strlen(infix);i++)
42     {
43         symbol=infix[i];
44         while (F(s[top])>G(symbol))
45         {
46             postfix[j]=s[top-];
47             j++;
48         }
49         if (F(s[top])!=G(symbol))
50             s[++top]=symbol;
51         else
52             top--;
53     }
```

15:422.8KB/s4G LTE

new*

```
54 while (s[top]!='#')
55 {
56     postfix[j++] = s[top--];
57 }
58 postfix[j] = '\0';
59 }
60 int main()
61 {
62     char infix [20];
63     char postfix [20];
64     printf ("enter the valid infix expression\n");
65     scanf ("%s",infix);
66     infix_postfix(infix,postfix);
67     printf ("the postfix expression is\n");
68     printf ("%s\n",postfix);
69 }
70
```

| < | > | (|) | [|]

Output:

```
15:40 0.0KB/s 4G LTE 9
← TAB _ ⋮
enter the valid infix expression
a*b*c-d+e/f/(g+h)
the postfix expression is
ab*c*d-ef/gh+/*
[Program finished]
```

: