

~\Desk...(JAVA)
Stude...n.java
~\Desk...w folder
arr.c
fac.c
n.c
ne.c
p.c
post.c
pref.c

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

Status gcc -Wall -o "p" "p.c" (in directory: C:\Users\DELL\Desktop\New folder)

Compiler Compilation finished successfully.

Messages

Scribble

~\Desk...(JAVA)
Stude...n.java
~\Desk...w folder
arr.c
fac.c
n.c
ne.c
p.c
post.c
pref.c

```
37         return 5;
38     case '(':
39         return 0;
40     case ')':
41         return 9;
42     default:
43         return 7;
44     }
45 }
46 void infix_prefix(char infix[], char prefix[])
47 {
48     int top, j, i;
49     char s[30], symbol;
50     top = -1;
51     s[++top] = '#';
52     j = 0;
53     strrev(infix);
54     for (i = 0; i < strlen(infix); i++)
55     {
56         symbol = infix[i];
57         while (F(s[top]) > G(symbol))
58         {
59             prefix[j] = s[top--];
60             j++;
61         }
62         if (F(s[top]) != G(symbol))
63         {
64             s[++top] = symbol;
65         }
66         else
67         {
68             top--;
69         }
70     }
71     while (s[top] != '#')
72     {
73         prefix[j++] = s[top--];
74     }
```

Status gcc -Wall -o "p" "p.c" (in directory: C:\Users\DELL\Desktop\New folder)
Compiler Compilation finished successfully.

Messages

Scribble


```

70     }
71     while (s[top] != '#')
72     {
73         prefix[j++] = s[top--];
74     }
75     prefix[j] = '\0';
76     strrev(prefix);
77 }
78
79 int main()
80 {
81     char infix[30], prefix[30];
82     printf("Enter the valid infix expression:\n");
83     scanf("%s", infix);
84     infix_prefix(infix, prefix);
85     printf("The prefix expression is:\n");
86     printf("%s\n", prefix);
87     return 0;
88 }
89
90

```

Status gcc -Wall -o "p" "p.c" (in directory: C:\Users\DELL\Desktop\New folder)

Compiler Compilation finished successfully.

Messages

Scribble

not 07/1/20

Enter the valid infix expression:

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

The prefix expression is:

+-*^abcd//ef+gh

(program exited with code: 0)

Press any key to continue . . .