

```

#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]='#';
    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';
}

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

```

```

main.c
1
2 #include <stdio.h>
3 #include <string.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 '$': return 5;
15            case '(': return 0;
16            case '#': return -1;
17            default: return 8;
18        }
19    }
20    int G(char symbol)
21    {
22        switch(symbol)
23        {
24            case '+':
25            case '-': return 1;
26            case '*':
27            case '/': return 3;
28            case '^':
29            case '$': return 6;
30            case '(': return 9;
31            case ')': return 0;
32            default: return 7;
33        }
34    }
35    void infix_postfix(char infix[], char postfix[])
36    {
37        int top,i,j;
38        char s[30],symbol;
39        top=-1;
40        s[++top]='#';
41        j=0;
42        for(i=0;i<strlen(infix);i++)
43        {
44            symbol=infix[i];
45            while(F(s[top])>G(symbol))
46            {
47                postfix[j]=s[top--];
48                j++;
49            }
50            if(F(s[top])!=G(symbol))
51                s[++top]=symbol;
52            else
53                top--;
54        }
55        while(s[top]!='#')
56        {
57            postfix[j++]=s[top--];
58        }
59        postfix[j]='\0';
60    }
61    void main()
62    {
63        char infix[20];
64        char postfix[20];
65        printf("Enter the valid infix expression\n");
66        scanf("%s",infix);
67        infix_postfix(infix,postfix);
68        printf("The postfix exp is \n");
69        printf("%s\n",postfix);
70    }
71 }

```

Input

```

Enter the valid infix expression
a+b*(c^d-e)^(f+g*h)-i
The postfix exp is
abcd^e-fgh*+^*+i-

```

```

...Program finished with exit code 0
Press ENTER to exit console.

```