

Code:

```
19  #include <stdio.h>
20  #include <stdlib.h>
21  #include <ctype.h>
22  #include <string.h>
23  #define MAX 50
24
25  char stack[MAX];
26  int top = -1;
27
28  void push(char c)
29  {
30      if(top == MAX-1)
31          printf("Stack overflow!!!");
32      else
33      {
34          top++;
35          stack[top] = c;
36      }
37  }
38
39  char pop()
40  {
41      char c;
42      if(top == -1)
43          return -1;
44      else
45      {
46          c = stack[top];
47          top--;
48          return c;
49      }
50  }
51
52  int isOperator(char c)
53  {
54      if(c == '^' || c == '*' || c == '/' || c == '+' || c == '-')
55          return 1;
56      else
57          return 0;
58  }
59
60  int precedence(char c)
61  {
```

```

62     if(c == '^')
63         return 3;
64     else if(c == '*' || c == '/')
65         return 2;
66     else if(c == '+' || c == '-')
67         return 1;
68     else
69         return 0;
70 }
71
72 void convert(char infix[],char postfix[])
73 {
74     int i=0,j=0;
75     char item,x;
76
77     push('(');
78     strcat(infix,"");
79
80     item = infix[i];
81
82     while(item != '\0')
83     {
84         if(item == '(')
85             push(item);
86         else if(isalpha(item) || isdigit(item))
87         {
88             postfix[j] = item;
89             j++;
90         }
91         else if(isOperator(item) == 1)
92         {
93             x = pop();
94             while(isOperator(x) == 1 && precedence(x) >= precedence(item))
95             {
96                 postfix[j] = x;
97                 j++;
98                 x = pop();
99             }
100             push(x);
101             push(item);
102         }
103         else if(item == ')')
104         {

```

```

105         x = pop();
106         while(x != '(')
107         {
108             postfix[j] = x;
109             j++;
110             x = pop();
111         }
112     }
113     else
114     {
115         printf("Invalid expression!\n");
116         exit(0);
117     }
118     i++;
119     item = infix[i];
120 }
121
122 if(top > 0)
123 {
124     printf("Invalid expression!!!\n");
125     exit(0);
126 }
127 postfix[j] = '\0';
128 }
129
130 int main(int argc, char **argv)
131 {
132     char infix[MAX], postfix[MAX];
133
134     printf("Enter the infix expression: ");
135     gets(infix);
136     convert(infix, postfix);
137     printf("\nYour postfix expression is:");
138     puts(postfix);
139     return 0;
140 }

```