

main.c

```
1 #include<stdio.h>
2 #define max 100
3 char stack[max];
4 int top = -1;
5
6 void push(char ch)
7 {
8     if (top == max - 1)
9         printf("Stack is full\n");
10    else
11    {
12        top++;
13        stack[top] = ch;
14    }
15 }
16 char pop()
17 {
18     char ch=-1;
19     if (top == -1)
20         printf("stack is empty\n");
21     else
22     {
23         ch = stack[top];
24         top--;
25     }
26 }
27 return ch;
28 }
```

main.c

```
--  
27     return ch;  
28 }  
29 int stackempty()  
30 {  
31     if (top == -1)  
32         return 1;  
33     return 0;  
34 }  
35 char stacktop()  
36 {  
37     if (top == -1)  
38         return -1;  
39     else  
40     {  
41         char ch = stack[top];  
42         return ch;  
43     }  
44 }  
45 int priority(char ch)  
46 {  
47     if (ch == '^')  
48         return 3;  
49     else if (ch == '/')  
50         return 2;  
51     else if (ch == '*')  
52         return 2;  
53     else if (ch == '-')  
54         return 1;
```

```
-- -----
51     else if (ch == '*')
52         return 2;
53     else if (ch == '-')
54         return 1;
55     else if (ch == '+')
56         return 1;
57     else
58         return 0;
59
60 }
61 int main()
62 {
63     int i,k=0;
64     char infix[100], postfix[100];
65     printf("Enter the infix expression : ");
66     scanf("%s", infix);
67     printf("The postfix expression is : \n");
68     for (i = 0; infix[i] != '\0'; i++)
69     {
70         if (infix[i] == '(')
71         {
72             push(infix[i]);
73         }
74         else if (infix[i] == ')')
75         {
76             while (stacktop() != '(')
77             {
78                 pop();
79                 push(stacktop());
80             }
81             pop();
82         }
83     }
84 }
```

```
1     while (stacktop() != '(')
2     {
3         postfix[k] = pop();
4         k++;
5     }
6     pop();
7 }
8 else if (infix[i] == '^' || infix[i] == '/' || infix
9 [i] == '*' || infix[i] == '-' || infix[i] == '+')
10 {
11     while(stackempty() == 0 && priority(stacktop()) >=
12 priority(infix[i]))
13     {
14         postfix[k] = pop();
15         k++;
16     }
17     push(infix[i]);
18 }
19 else
20 {
21     postfix[k] = infix[i];
22     k++;
23 }
24 }
25     int stackempty()
26 while(!stackempty())
27 {
```

main.c

```
100     {
101         postfix[k] = pop();
102         k++;
103     }
104     for(i=0;i<k;i++)
105     {
106         printf("%c",postfix[i]);
107     }
108 }
```

<https://LovelyEnchantingMetric.ashutoshupadhy1.repl.run>

```
► clang-7 -pthread -lm -o main main.c
► ./main
Enter the infix expression : a+b-(c*d/e)
The postfix expression is :
ab+cd*e/-> []
```