

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 item;
19     if (top== -1)
20         printf("\n stack is empty !");
21     else
22     {
23         item=stack[top];
24         top--;
25         return item;
26     }
27 }
28
29
30 int stackempty()
31 {
32     if(top== -1) return 1;
```

main.c

```
30 int stackempty()
31 {
32     if(top==-1) return 1;
33     else return 0;
34 }
35
36 char stacktop()
37 {
38     if( top==-1)
39         printf("\n stack is empty!");
40     else
41         return stack[top];
42 }
43 int priority(char ch)
44 {
45     switch(ch)
46     {
47         case '+':
48         case '-':return (1);
49         case '*':
50         case '/':return (2);
51         case '^': return (3);
52         default : return (0);
53     }
54 }
55
56
57
58 int main(int argc, char **argv)
59 {
60     char infix[100];
61     int i, item;
62     printf("Enter the infix expression :");
```



main.c

```
60 char infix[100];
61 int i, item;
62 printf("Enter the infix expression :");
63 scanf("%s",infix);
64 printf("Expression : %s",infix);
65 printf("\n Postfix: ");
66 i=0;
67 while (infix[i]!='\0')
68 {
69
70
71     switch (infix[i])
72     {
73         case '(': push(infix[i]);
74                 break;
75         case ')': while(( item=pop())!='(')
76                 printf("%c",item);
77                 break;
78         case '+':
79         case '-':
80         case '*':
81         case '/':
82         case '^':
83             while(!stackempty() && priority(infix[i])<=priority(stacktop()))
84             {
85                 item=pop();
86
87                 printf("%c", item);
88             }
89             push(infix[i]);
90
```

main.c

```
82         case '^':
83             while(!stackempty() && priority(infix[i])<=priority(stacktop()))
84                 {
85                     item=pop();
86                     printf("%c", item);
87                 }
88             push(infix[i]);
89             break;
90         default : printf("%c", infix[i]);
91                 break;
92     }
93     i++;
94 }
95
96 while(!stackempty())
97 {
98     char item;
99     item=pop();
100     printf("%c", item);
101 }
102 printf("\n");
103 return 0;
104
105 }
106 }
```