

```
1  #include<stdio.h>
2  #include<ctype.h>
3  #define SIZE 50
4  char stack[SIZE];
5  int top=-1;
6  push(char elem)
7  {
8  stack[++top]=elem;
9  }
10 char pop()
11 {
12 return(stack[top--]);
13 }
14 int pr(char symbol)
15 {
16 if(symbol== '^')
17 {
18 return(3);
19 }
20 else if(symbol== '*' || symbol== '/' )
21 {
22 return(2);
23 }
24 else if(symbol== '+' || symbol== '-')
25 {
26 return(1);
27 }
28 else {
29 return(0);
30 }
31 }
32 void main()
33 {
```

```

29     return(0);
30 }
31 }
32 void main()
33 {
34     char infix[50], postfix[50], ch, elem;
35     int i=0, k=0;
36     printf("enter Infix expression");
37     scanf("%s", infix);
38     push('#');
39     while((ch=infix[i++])!='\0' )
40     {
41         if(ch == '(') push(ch);
42         else
43             if(isalnum(ch)) postfix[k++]=ch;
44         else
45             if(ch==')')
46             { while(stack[top] != '(')
47                 postfix[k++]=pop();
48                 elem=pop();
49             }
50             else
51             {
52                 while(pr(stack[top]) >= pr(ch))
53                     postfix[k++]=pop();
54                 push(ch);
55             }
56             }
57         while(stack[top]!='#')
58             postfix[k++]=pop();
59         postfix[k]='\0';
60         printf("\nPostfixexpression=%s\n", postfix);
61     }

```





enter Infix expression4+6\*2

Postfixexpression=462\*+

...Program finished with exit code 0

press ENTER to exit console.