

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #define MAXSIZE 100
5
6  typedef union
7  {
8      float fData;
9      char cData;
10 } Item;
11
12
13 typedef struct
14 {
15     Item items[MAXSIZE];
16     int top;
17 } Stack;
18
19 int Priority (char *c)
20 {
21     if(c=='^')return 3;
22     else if (c=='*' || c=='/')return 2;
23     else if (c=='+' || c=='-')return 1;
24     else return 0;
25 }
26 Stack* initialize()
27 {
28     Stack *s = malloc(sizeof (Stack));
29     s->top=0;
30     return s;
31 }
32
33
34 Item pop (Stack *s)
35 {
36     return s->items[--s->top];
37 }
38
39
40 void push(Stack *s,Item value)
41 {
42     s->items[s->top++]=value;
43 }
44
45
46 Item top (Stack *s)
47 {
48     return s->items[--s->top];
49 }
50
51
52 int isEmpty(Stack *s)
53 {
54     if(s->top==0) return 1;
55     else return 0;
56 }
57
58 void infixTopostfix(char *infix, char* postfix)
59 {
60     int i,j=0;
61     Item x;
62     Stack *p = initialize();
63     *postfix="";
64     for(i=0;i<strlen(infix);i++)
65     {
66         if(infix[i]!=' ')continue ;

```

```

67     if(isdigit(infix[i]) || isalpha(infix[i])){
68         strcat(postfix,infix[i],1);
69     }
70     else if (infix[i]=='(')
71     {
72         x.cData=infix[i];
73         push(p,x);
74     }
75     else if (infix[i]==')')
76     {
77         while (top(p).cData!='(')
78             strcat(postfix , pop(p).cData,1);
79         pop(p);
80     }
81
82     else
83     {
84         while (!isEmpty(p) && (Priority(infix[i]) <= Priority(top(p).cData)))
85             strcat(postfix, pop(p).cData,1);
86
87         x.cData = infix[i];
88         push(p,x);
89     }
90 }
91
92 }
93
94 }
95 int main()
96 {
97     Stack *s = initialize();
98     char *infix = (char*)malloc(100);
99     strcpy(infix,"1 + 2 + 3 * 4");
100    char *postfix = (char*)malloc(100);
101    infixTopostfix(infix,postfix);
102    printf("%s",postfix);
103    return 0;
104 }

```