

Name: Shree Ganesh R Vishwakarma

Class: SYIT

Roll No. 65

```
shree1.c
~/Desktop

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <ctype.h>
4 #include <string.h>
5 const int g_max_size=100;
6 char st[100];
7 int top = -1;
8 void push(char item){
9     if(top==g_max_size-1){
10         printf("\nstack overflow\n");
11     }
12     else{
13         ++top;
14         st
15         [top]=item;
16     }
17 }
18 char pop(){
19     char item;
20     if(top==0){
21         printf("\nstack underflow:invalid infix\n");
22         printf("exitlag....\n");
23         exit(1);
24     }
25     else{
26         item=st
27         [top];
28         --top;
29         return item;
30     }
31 }
32
33 int is_operator(char symbol){
34     if(symbol=='+'||symbol=='*'||symbol=='/'||symbol=='%'){
35         return 1; //is an operator, true
36     }
37     else{
38         return 0; //not an operator, false
39     }
40 }
41 int precedence(char symbol){
42     switch(symbol)
43     {
44     case '^': return 3;
45
46     case '*': return 2;
47
48     case '/': return 2;
49
50     case '+': return 1;
51
52     case '-': return 1;
53
54     default: return 0;
55 }
56 return 0;
57 }
58
59 void infixToPostfix(char infix_exp[],char postfix_exp[]){
60     int i=0;
61     int j=0;
62     char item;
63     char x;
64     push('(');
65     strcat(infix_exp,"");
66     item=infix_exp[i];
67     while(item!='\0'){
68         if(item!='('){
69             push(item);
70         }
71         else if(!isdigit(item)||!isalpha(item)){
72             postfix_exp[j]=item;
73             j++;
74         }
75         else if(is_operator(item)==1){
76             x=pop();
77             while(is_operator(x)==1&&precedence(x)>=precedence(item)){
78                 postfix_exp[j]=x;
79                 j++;
80             }
81             x=pop();
82             push(x);
83             push(item);
84         }
85         else if(item==')'){
86             x=pop();
87             while(x!='('){
88                 postfix_exp[j]=x;
89                 j++;
90             }
91             x=pop();
92             postfix_exp[j]=x;
93             j++;
94         }
95     }
96     postfix_exp[j]='\0';
97 }
```

```
Open shree1.c Save
71 while (x!=0 && isoperator(x)) { x=pop(); continue; }
72 postfix_exp[j]=item;
73 j++;
74 }
75 else if (is_operator(item)==1){
76 x=pop();
77 while (is_operator(x)==1 && precedence(x)>=precedence(item)){
78 postfix_exp[j]=x;
79 j++;
80 x=pop();
81 }
82 push(x);
83 push(item);
84 }
85 else if (item==' '){
86 x=pop();
87 while (x!='('){
88 postfix_exp[j]=x;
89 j++;
90 x=pop();
91 }
92 }
93 else{
94 printf("\nInvalid infix expression\n");
95 getchar();
96 exit(1);
97 }
98 i++;
99 item=infix_exp[i];
100 }
101 postfix_exp[j]='\0';
102 }
103
104
105 int main()
106 {
107 char infix[100];
108 char postfix[100];
109 printf("Enter infix expression: ");
110 scanf("%s",infix);
111 infixToPostfix(infix, postfix);
112 printf("Postfix expression:%s\n",postfix);
113 return 0;
114 }
115

C Tab Width: 8 Ln 115, Col 1 INS
di404@itadmin: ~/Desktop
di404@itadmin:~/Desktop$ gcc shree1.c
di404@itadmin:~/Desktop$ ./a.out
Enter infix expression: A+B
Postfix expression:AB+
di404@itadmin:~/Desktop$
```