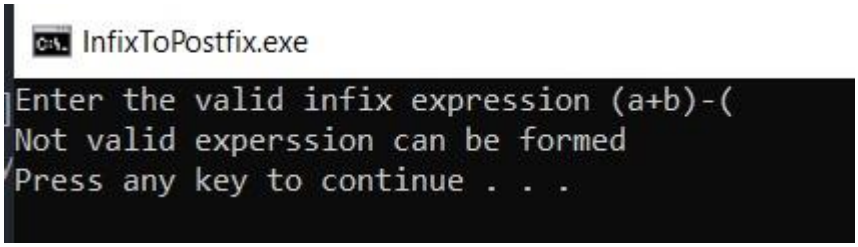
```
//LAB 2 INFIX TO POSTFIX CONVERSION
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int F(char symbol)
{
switch(symbol)
{
 case'+':
 case'-':return 2;
 case'*':
 case'/':return 4;
 case'^':
 case'$':return 5;
 case'(':return 0;
 case'#':return -1;
 default:return 8;
 }
}

int G(char symbol)
{
switch(symbol)
  {
 case'+':
 case'-':return 1;
 case'*':
 case'/':return 3;
 case'^':
 case'$':return 6;
 case'(':return 9;
```

```
 case')':return 0;
 default:return 7;
  }
}


int infix_postfix(char infix[],char postfix[])
{
int top,i,j,d=0,f=0;
char s[30],symbol;
top=-1;
s[++top]='#';
j=0;

for(i=0;i<strlen(infix);i++)
 {
   if(infix[i]=='('){
   d++;}
   else if (infix[i]==')')
   f++;
symbol=infix[i];
while(F(s[top])>G(symbol))
{
postfix[j]=s[top--];
j++;
}


if(F(s[top])!=G(symbol))
s[++top]=symbol;
else
```

```
top--;
}

while(s[top]!='#')
{
postfix[j++]=s[top--];
}
postfix[j]='\0';
return (d+f);
}

void main()
{int a;
char infix[20];
char postfix[20];
printf("Enter the valid infix expression ");
scanf("%s",infix);
a= infix_postfix(infix , postfix );
if((strlen(postfix)+a)!=strlen(infix))
printf("Not valid experssion can be formed \n");
else
printf("The postfix expression is :\t%s\n",postfix);

}
```



```
Enter the valid infix expression
(A+(B-C)*D)
The postfix expresssion is ABC-D*+
```