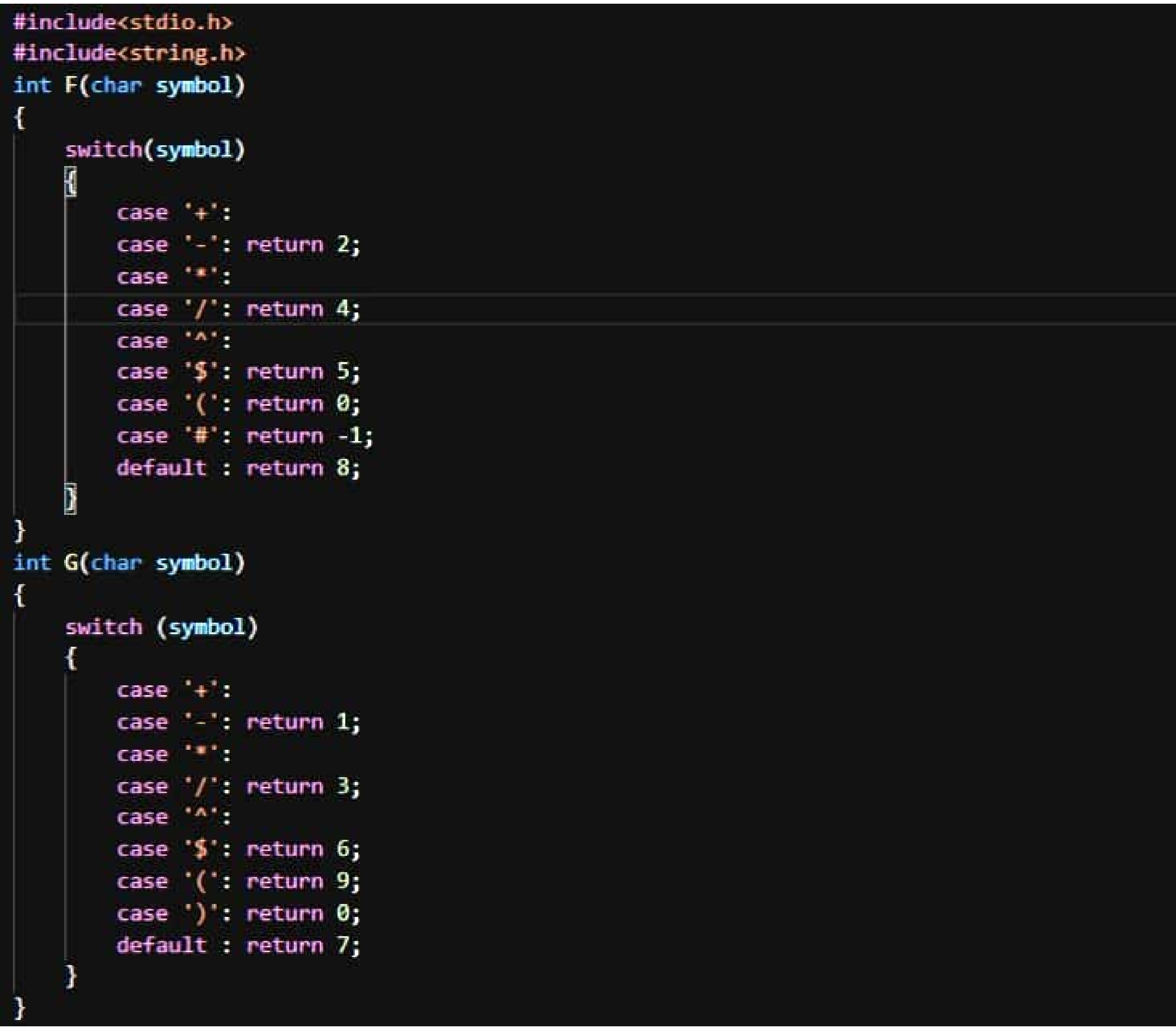
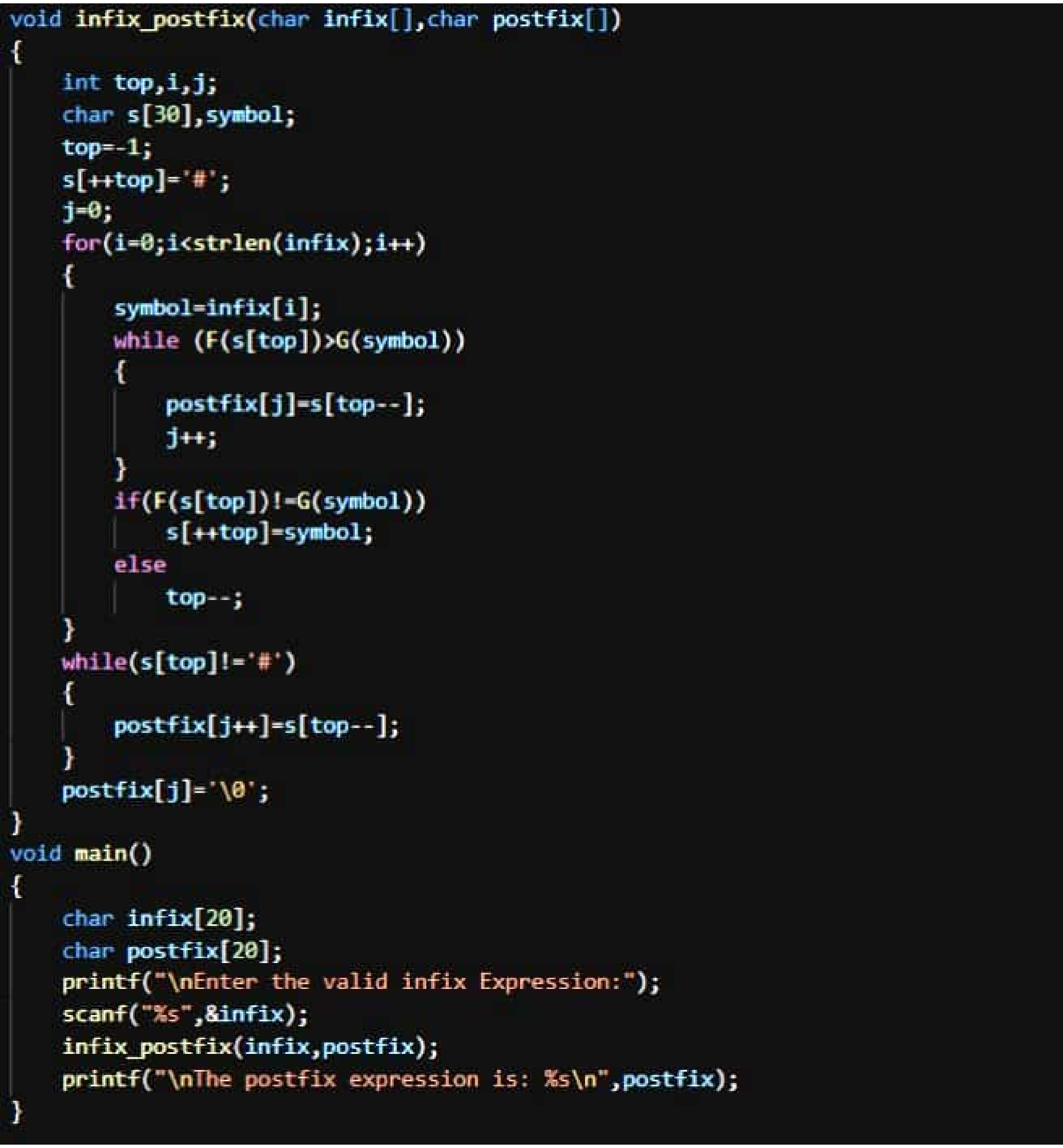
Lab program 2:-

```
#include<stdio.h>
#include<string.h>
int F (char symbol)
{
   switch(symbol)
   {
     case'+':
     case'-': return 2;
     case'*':
     case'/': return 4;
     case'^':
     case'$': return 5;
     case'(': return 0;
     case'#': return ~~0~~-1;
     default: return 8;
   }
}
int G (char symbol)
{
  switch (symbol)
  {
    case'+':
    case'-': return 1;
    case'*': ~~aa~~
    case'/': return 3;
    case'^':
    case'$': return 6;
    Case €'(': return 9;
    Case')': return 0;
    default: return 7;
  }
}
```

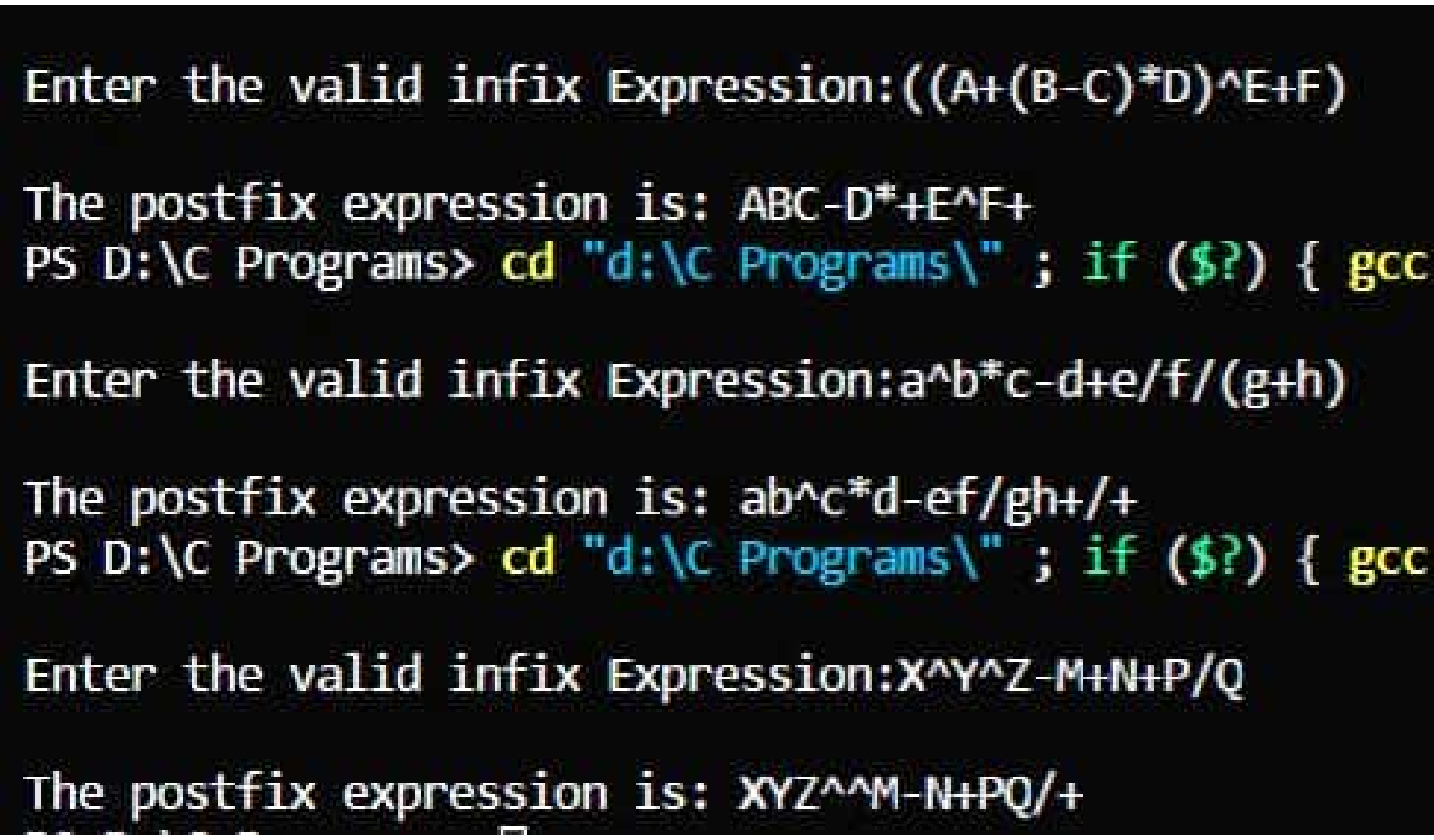
```
void infix_postfix(char infix[], char postfix[])
{
    int top,i,j;
    char s[30],symbol;
    top=-1;
    s[++top]='#';
    j=0;
    for(i=0; i<strlen(infix); i++)
    {
        symbol=infix[i];
        while (F(s[top])>G(symbol))
        {
            postfix[j]=s[top--];
            j++;
        }
        if(F(s[top])!=G(symbol))
            s[++top] =symbol;
        else
            top--;
    }
    while(s[top]!='#')
    {
        postfix[j++]=s[top--];
    }
    postfix[j]='\0';
}
void main()
{
    char infix[20];
    char postfix[20];
    printf("Enter the valid infix expression: ");
    scanf("%s",&infix);
    infix_postfix(infix,postfix);
    printf("The postfix expression is : %s \n",postfix);
}
```



```c
#include<stdio.h>
#include<string.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;
    }
}
```



```c
void infix_postfix(char infix[],char postfix[])
{
    int top,i,j;
    char s[30],symbol;
    top=-1;
    s[++top]='#';
    j=0;
    for(i=0;i<strlen(infix);i++)
    {
        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';
}
void main()
{
    char infix[20];
    char postfix[20];
    printf("\nEnter the valid infix Expression:");
    scanf("%s",&infix);
    infix_postfix(infix,postfix);
    printf("\nThe postfix expression is: %s\n",postfix);
}
```



```
Enter the valid infix Expression:((A+(B-C)*D)^E+F)

The postfix expression is: ABC-D*+E^F+
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc

Enter the valid infix Expression:a^b*c-d+e/f/(g+h)

The postfix expression is: ab^c*d-ef/gh+/+
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc

Enter the valid infix Expression:X^Y^Z-M+N+P/Q

The postfix expression is: XYZ^^M-N+PQ/+
```