```c
#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;
  }
}
```

```
17    default:return 8;
18    }
19  }
20
21  int G(char symbol)
22  {
23  switch(symbol)
24    {
25    case'+':
26    case'-':return 1;
27    case'*':
28    case'/':return 3;
29    case'^':
30    case'$':return 6;
31    case'(':return 9;
32    case')':return 0;
33    default:return 7;
34    }
35  }
36
```

Run code

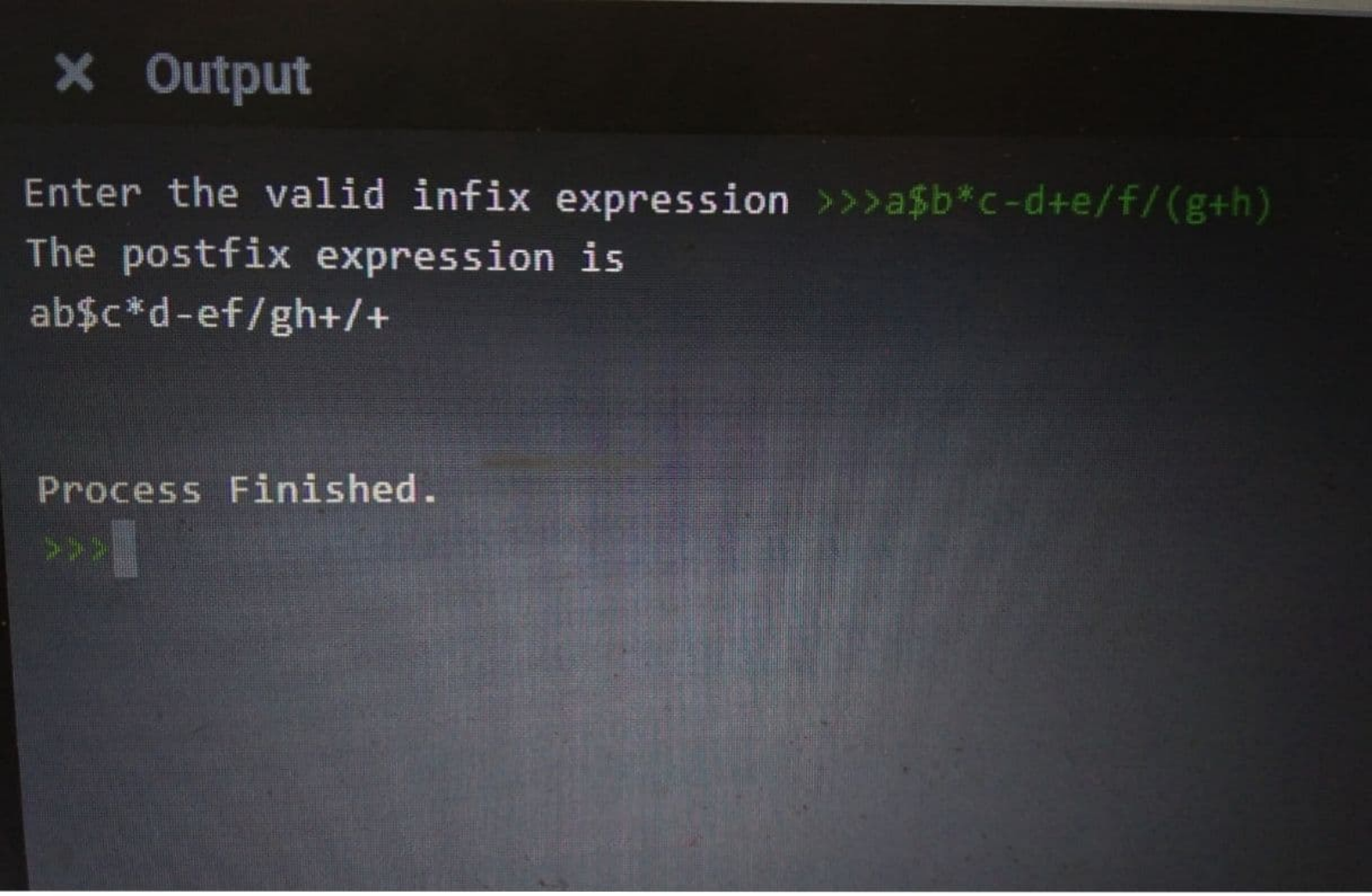
```
6
7  void infix_postfix(char infix[],char postfix[])
8  {
9  int top,i,j;
0  char s[30],symbol;
1  top=-1;
2  s[++top]='#';
3  j=0;
4
5  for(i=0;i<strlen(infix);i++)
6   {
7  symbol=infix[i];
8  while(F(s[top])>G(symbol))
9  {
0  postfix[j]=s[top--];
1  j++;
2  }
3
4
5  if(F(s[top])!=G(symbol))
```

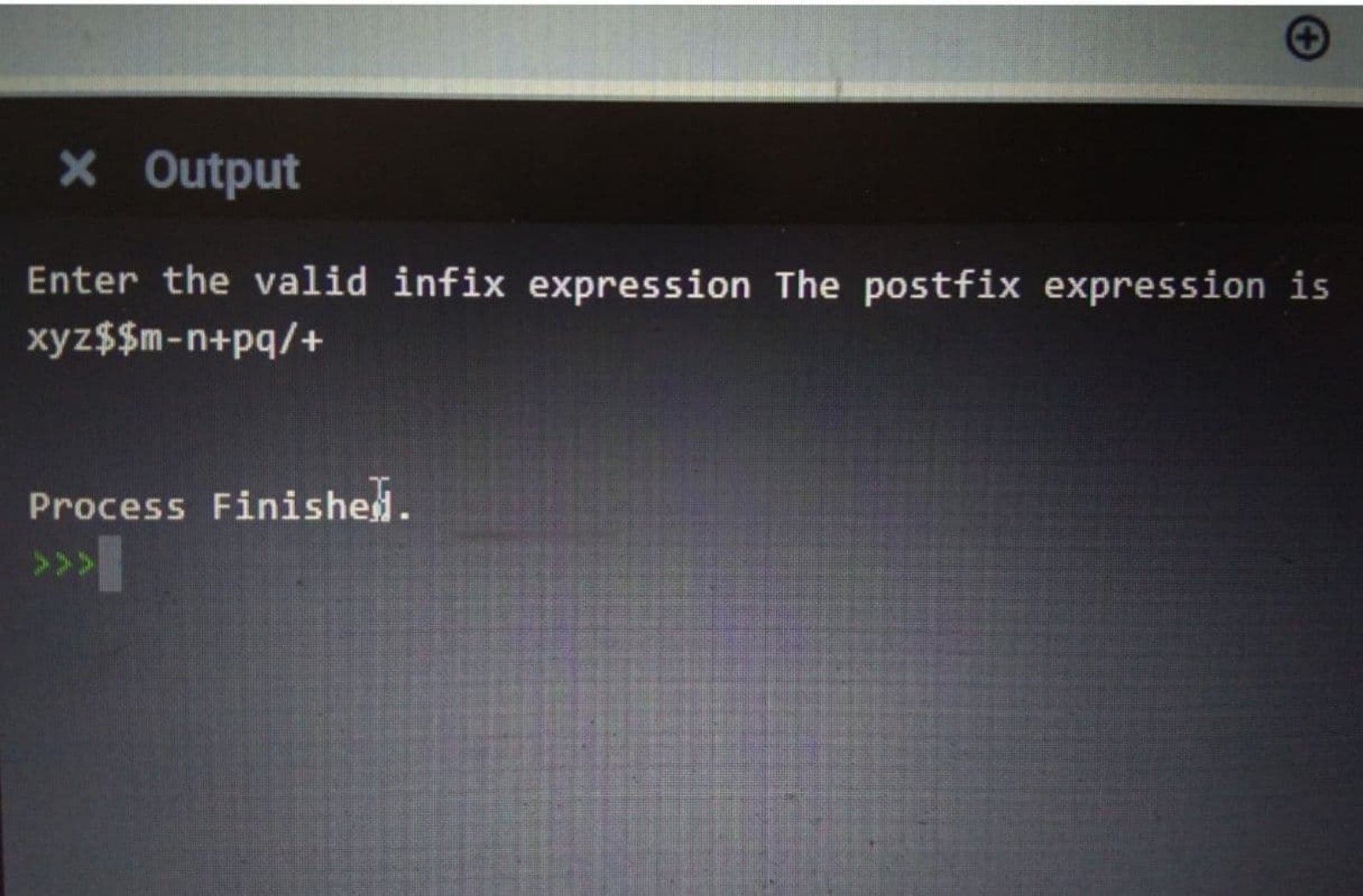
Run code

```
55  if(F(s[top])!=G(symbol))
56  s[++top]=symbol;
57  else
58  top--;
59  }
60
61  while(s[top]!='#')
62  {
63  postfix[j++]=s[top--];
64  }
65  postfix[j]='\0';
66  }
67
68  void main()
69  {
70  char infix[20];
71  char postfix[20];
72  printf("Enter the valid infix expression ");
73  scanf("%s",infix);
74  infix postfix(infix , postfix );
```

```
    }
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 \n");
printf("%s\n",postfix);
}
```


Output
Enter the valid infix expression >>>a$b*c-d+e/f/(g+h)
The postfix expression is
ab$c*d-ef/gh+/+
Process Finished.
>>>


Output
Enter the valid infix expression The postfix expression is
xyz$$m-n+pq/+
Process Finished.
>>>

PAGE NO.
DATE: / /

LAB-2

Qn:- WAP to convert a given valid parinthesized infix arithematic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

① (A+(B-C)*D)
② A+B.
③ A^b*c-d+e/f/(g+h).
④ xyz-m+n+p/q
⑤ a$b*c-d+e/f/(g+h)

Program:-

```
#include <stdio.h>
#include<string.h>
# include<stdlib.h>

int F(char symbol)
{
switch (symbol)
{
case '+':
case '-': retur 2;
cose'*':
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;
  }
}
```

```
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 \n");
printf("%s \n", postfix);
}
```