1.

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

int F(Sym char Symbol){
    Switch(Symbol){
    case '+':
    case '-': return 2;
    case '*':
    case '/': return 4;
    case '^':
    case '$': return 5;
    case '(': return 0;
    case '#': return -1;
    default: return 8;
}}
int G(Char Symbol){
    switch(Symbol){
    case'+':
    Cas'-': return 1;
    case'*':
    case'/': return 3;
    case'^':
    case'$': return 6;
    case'(': return 9;
    case')': return 0;
}}  default: return 7;
```

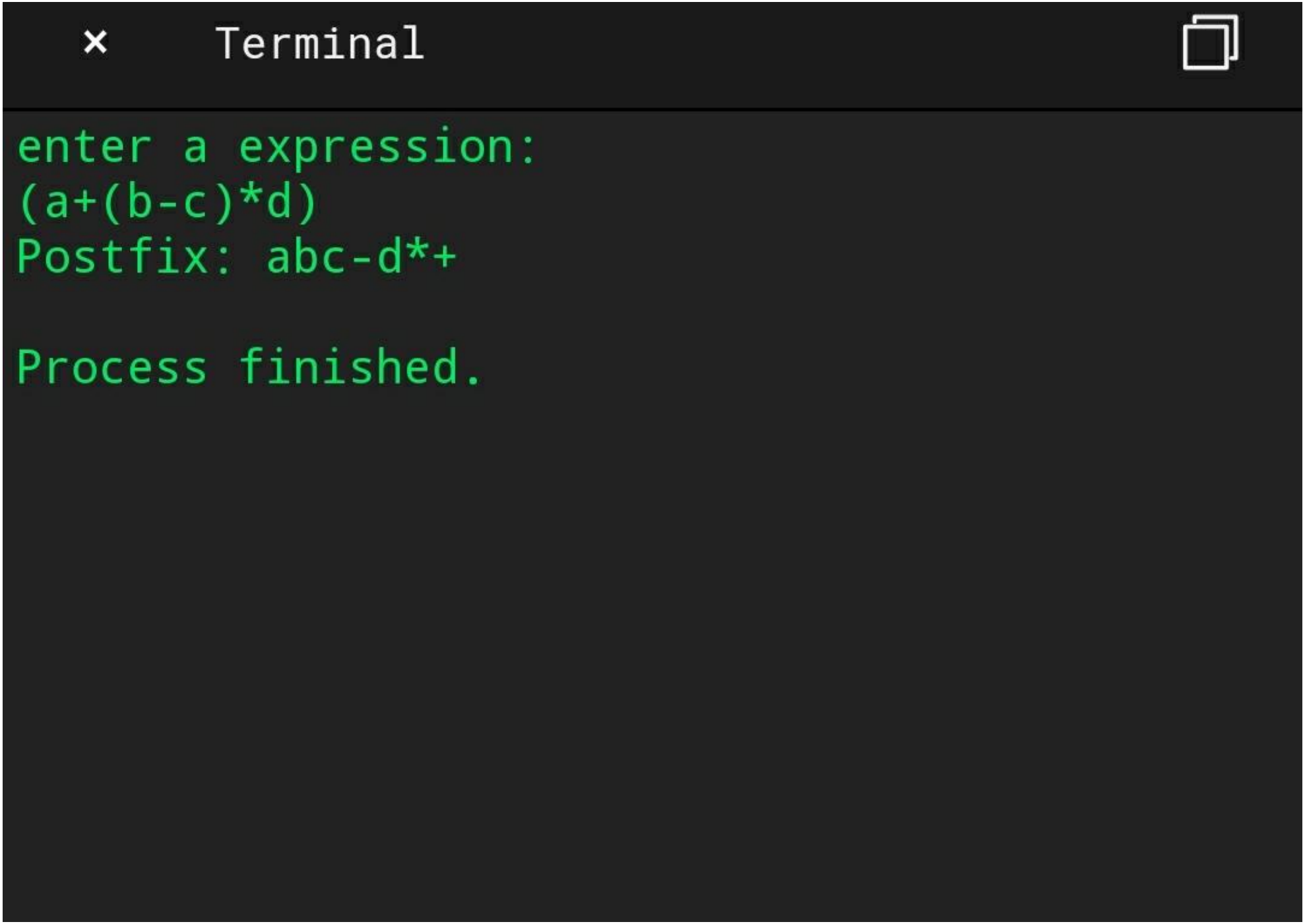
```
void infix_postfix(char infix[]){
    int top, j, i;
    char S[30], postfix[30];
    char symbol;
    top = -1;
    S[++top] = '#';
    j = 0;
    for(i=0; i < strlen(infix); i++){
        symbol = infix[i];
        while(F(S[top] > G(symbol)){
            postfix[j] = S[top--];
            j++;
        }
        if(F(S[top] != '#'){
            postfix[j++] =
            S[++top] = symbol;
        }
        else
            top--;
    }
    while(S[top] != '#'){
        postfix[j++] = S[top--];
    }
    postfix[j] = '\0';
    puts(postfix);
}
```

```
int main()
{
    char exp[30];
    printf("Enter an expression: \n");
    gets(exp);
    scanf("%s", exp);
    infix_Postfix(exp);
    return 0;
}
```

```c
#include<stdio.h>
#include<stdlib.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;
    }
}
void infix_postfix(char infix[]){
    int top,j,i;
    char s[30],postfix[30];
    char 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--;
    }
```

```
50      while(s[top]!='#'){
51          postfix[j++]=s[top--];
52      }
53      postfix[j]='\0';
54      printf("Postfix: ");
55      puts(postfix);
56  }
57  int main()
58  {
59      char exp[30];
60      printf("enter a expression:\n");
61      scanf("%s",exp);
62      infix_postfix(exp);
63      return 0;
64  }
```


Terminal
enter a expression:
(a+(b-c)*d)
Postfix: abc-d*+
Process finished.