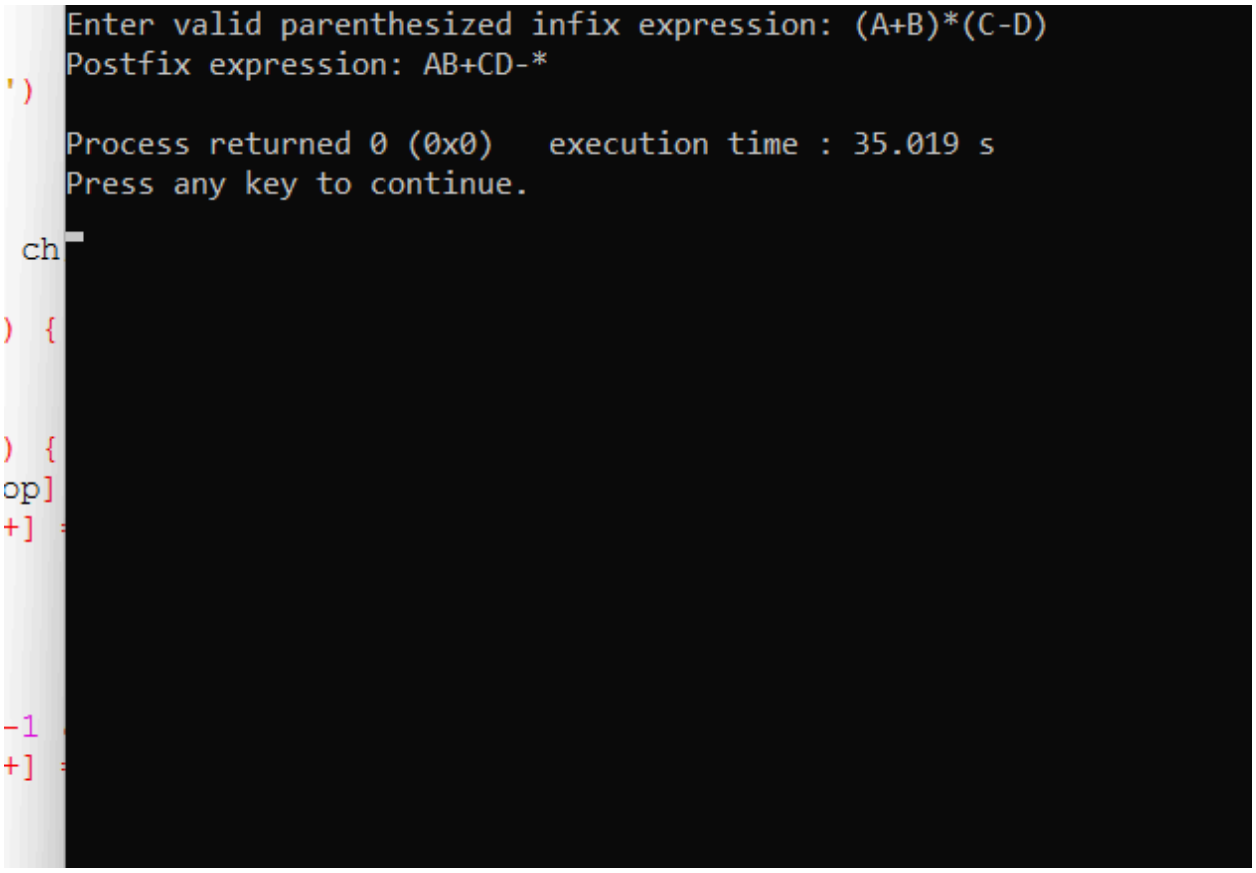
LAB_2 (infix to postfix)

```c
#include <stdio.h>
#include <ctype.h>
#define MAX 100

char stack[MAX];
int top = -1;
void push(char ch) {
    stack[++top] = ch;
}
char pop() {
    return stack[top--];
}
int precedence(char ch) {
    if (ch == '+' || ch == '-')
        return 1;
    if (ch == '*' || ch == '/')
        return 2;
    return 0;
}
int main() {
    char infix[MAX], postfix[MAX];
    int i = 0, k = 0;
    char ch;

    printf("Enter valid parenthesized infix expression: ");
    scanf("%s", infix);

    while (infix[i] != '\0') {
        ch = infix[i];

        if (isalnum(ch)) {
            postfix[k++] = ch;
        }
        else if (ch == '(') {
            push(ch);
        }
        else if (ch == ')') {
            while (stack[top] != '(') {
                postfix[k++] = pop();
            }
            pop();
        }
        else {
            while (top != -1 && precedence(stack[top]) >= precedence(ch)) {
                postfix[k++] = pop();
            }
            push(ch);
        }
        i++;
    }
    while (top != -1) {
        postfix[k++] = pop();
    }
    postfix[k] = '\0';
    printf("Postfix expression: %s\n", postfix);
    return 0;
}
```

OUTPUT:


Enter valid parenthesized infix expression: (A+B)*(C-D)
Postfix expression: AB+CD-*
Process returned 0 (0x0)   execution time : 35.019 s
Press any key to continue.