13.

#include <stdio.h>

#include <ctype.h>

char lookahead;

void match(char expected) {

if (lookahead == expected) lookahead = getchar();

else { printf("Syntax Error\n"); exit(1); }

}

void E() { T(); E\_prime(); }

void E\_prime() { if (lookahead == '+') { match('+'); T(); E\_prime(); } }

void T() { if (lookahead == 'i') { match('i'); match('n'); match('t'); } else if (lookahead == '(') { match('('); E(); match(')'); } }

int main() {

lookahead = getchar();

E();

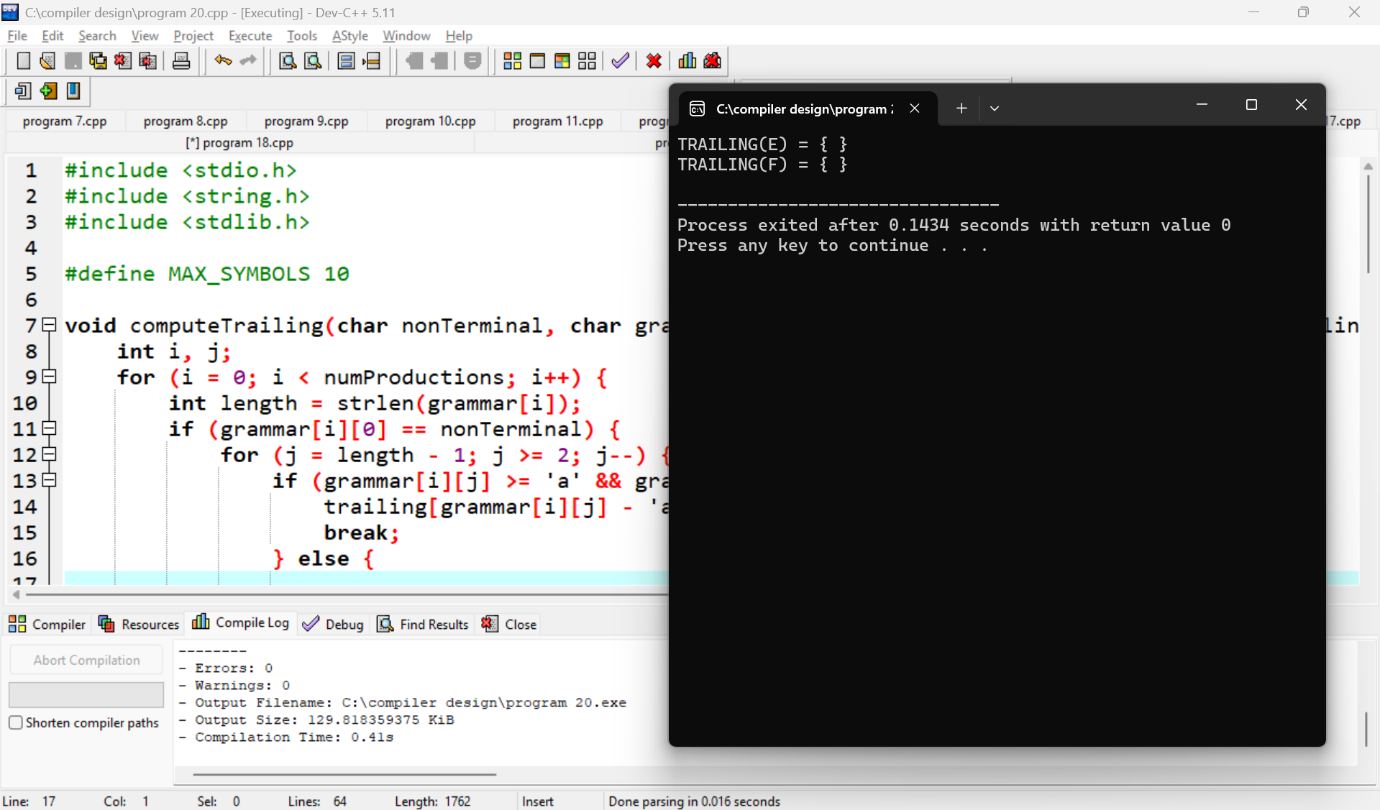
if (lookahead == '\n' || lookahead == EOF) printf("Parsing successful!\n");

else printf("Syntax Error\n");

return 0;

}

**Output:**

****

14.

#include <stdio.h>

#include <string.h>

#define MAX\_SIZE 100

char stack[MAX\_SIZE];

int top = -1;

void push(char c) {

if (top < MAX\_SIZE - 1) stack[++top] = c;

}

char pop() {

if (top >= 0) return stack[top--];

return -1;

}

int main() {

char input[] = "id+id\*id";

int i = 0;

while (input[i] != '\0') {

push(input[i]);

i++;

// Shift-Reduce Logic here

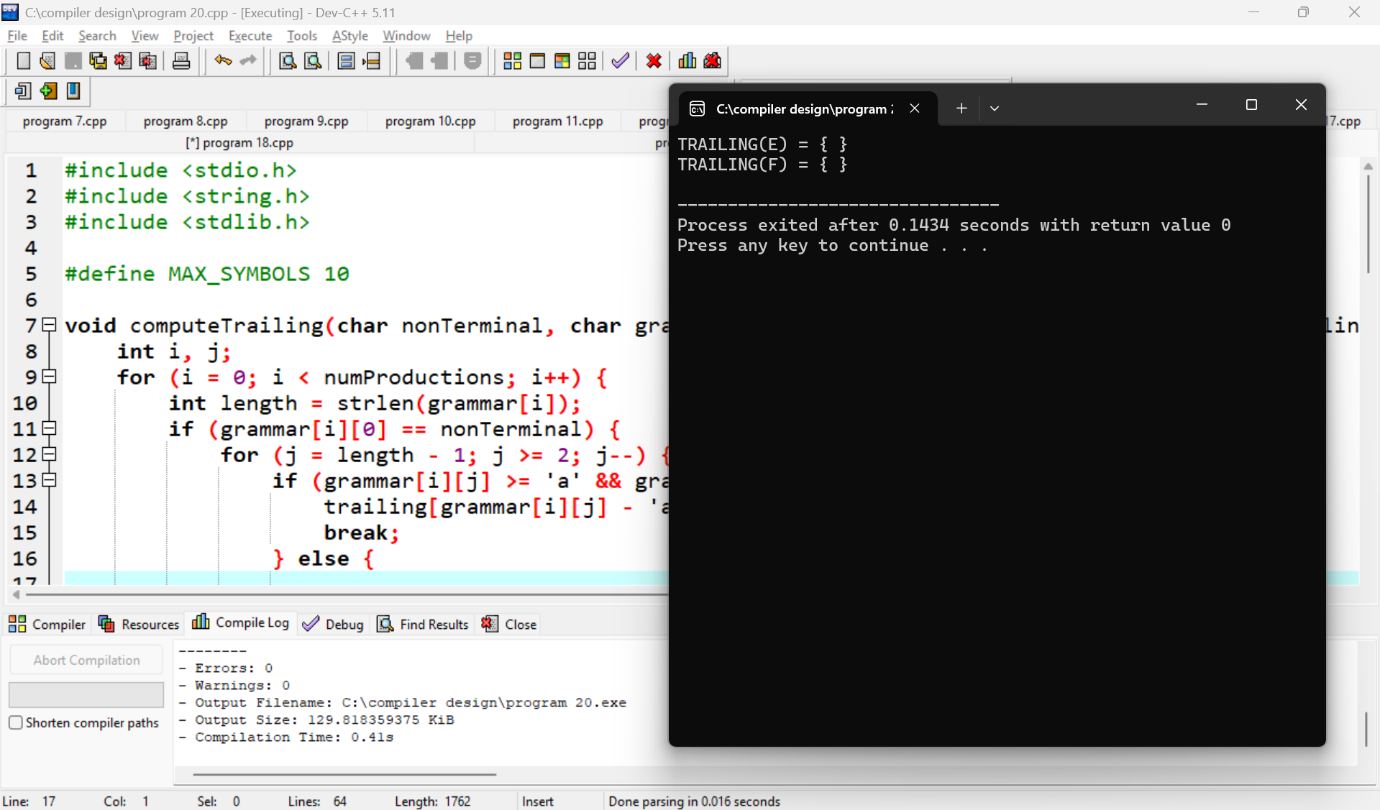
// For example, reduce id+id -> expression if the grammar allows it

}

return 0;

}

**Output:**

****