11.

#include <stdio.h>

#include <string.h>

#define MAX\_SYMBOLS 100

struct Symbol {

char name[20];

int value;

};

struct Symbol symbol\_table[MAX\_SYMBOLS];

int symbol\_count = 0;

void insert(char name[], int value) {

strcpy(symbol\_table[symbol\_count].name, name);

symbol\_table[symbol\_count].value = value;

symbol\_count++;

}

int lookup(char name[]) {

for (int i = 0; i < symbol\_count; i++) {

if (strcmp(symbol\_table[i].name, name) == 0)

return symbol\_table[i].value;

}

return -1; // Not found

}

int main() {

insert("x", 10);

insert("y", 20);

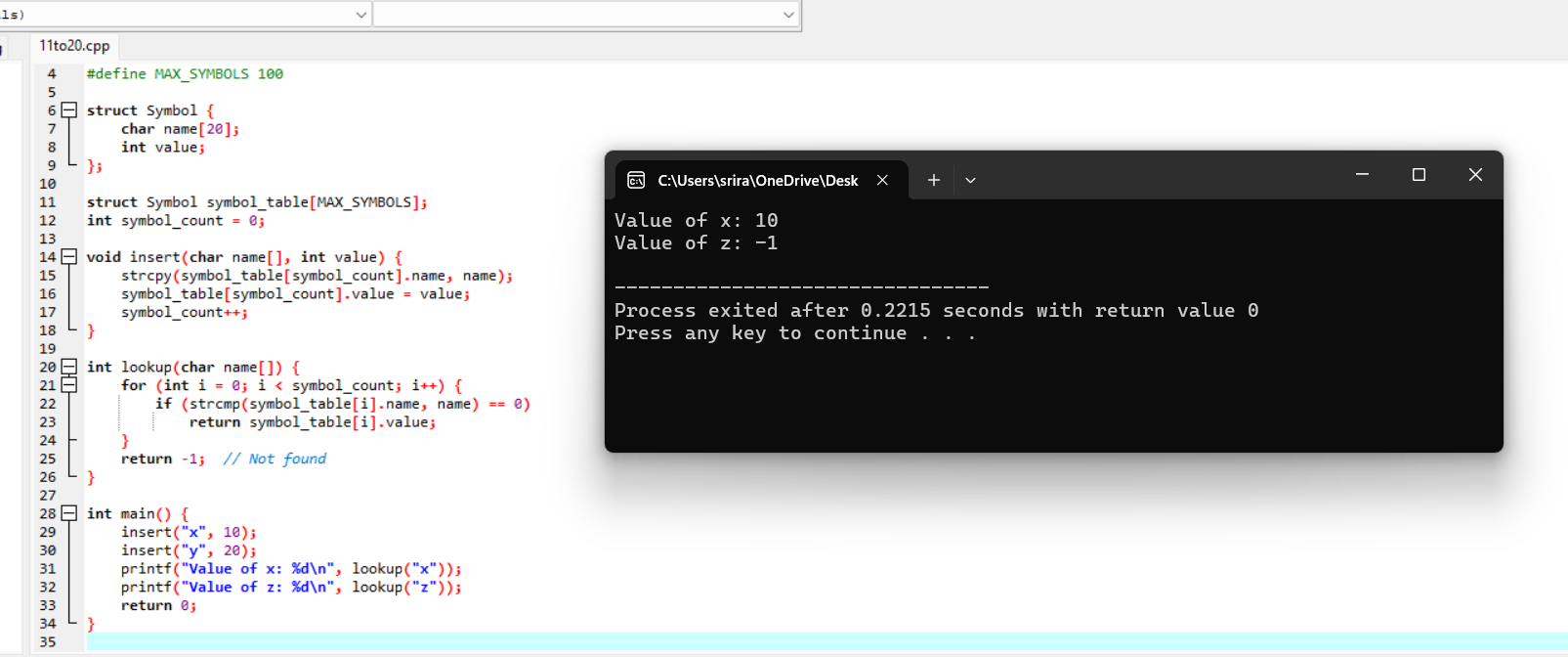
printf("Value of x: %d\n", lookup("x"));

printf("Value of z: %d\n", lookup("z"));

return 0;

}

**Output:**

****

12.

#include <stdio.h>

#include <ctype.h>

void expr();

void term();

void factor();

int lookahead;

void match(int t) {

if (lookahead == t) lookahead = getchar();

else printf("Syntax error\n");

}

void expr() {

term();

while (lookahead == '+' || lookahead == '-') {

match(lookahead);

term();

}

}

void term() {

factor();

while (lookahead == '\*' || lookahead == '/') {

match(lookahead);

factor();

}

}

void factor() {

if (isdigit(lookahead)) {

match(lookahead);

} else if (lookahead == '(') {

match('(');

expr();

match(')');

} else {

printf("Syntax error\n");

}

}

int main() {

lookahead = getchar();

expr();

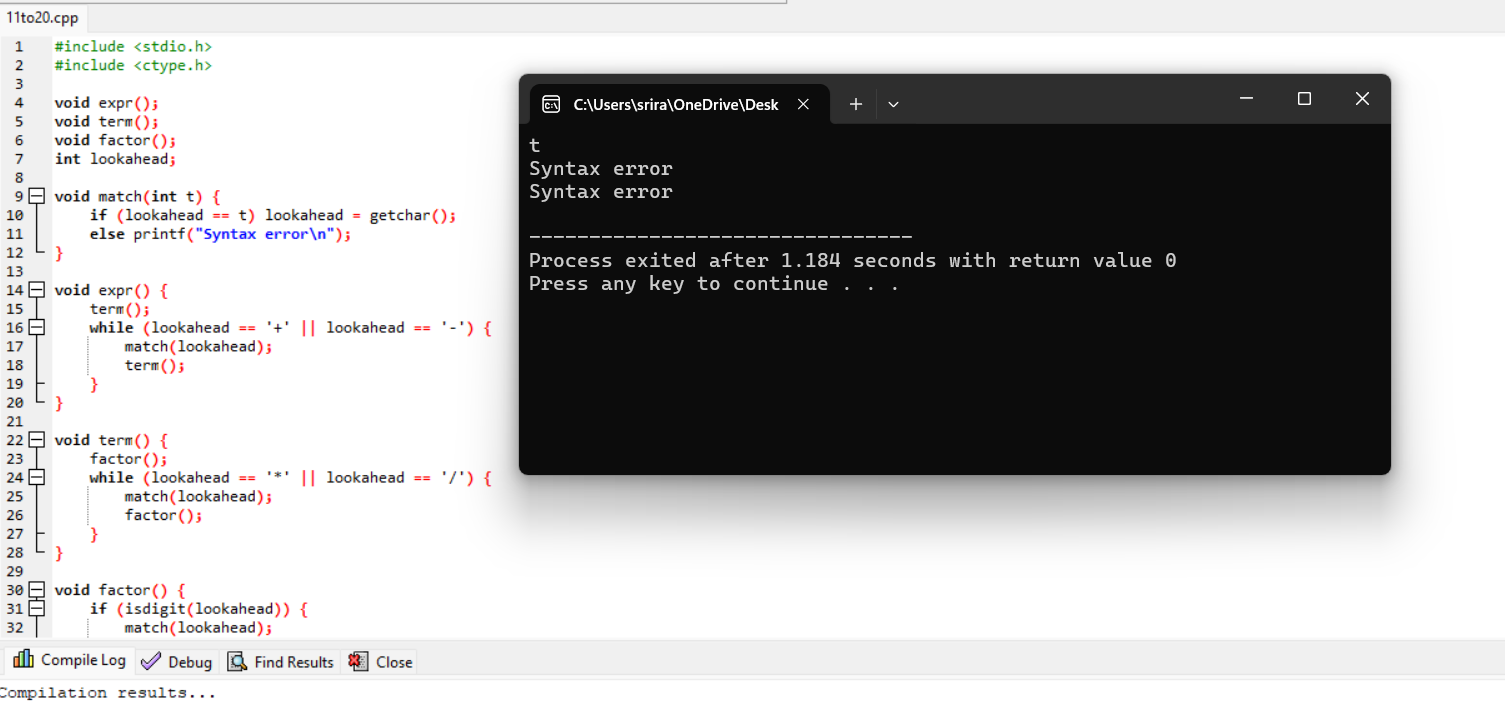
if (lookahead == EOF) printf("Parsing successful\n");

else printf("Syntax error\n");

return 0;

}

**Output:**

****