12. Write a C program to construct recursive descent parsing for the given grammar.

**Program:**

#include <stdio.h>

#include <ctype.h>

char\* input;

char lookahead;

void E();

void E\_();

void T();

void T\_();

void F();

void match(char expected) {

if (lookahead == expected) {

lookahead = \*input++;

} else {

printf("Syntax error: expected %c\n", expected);

exit(1);

}

}

void E() {

T();

E\_();

}

void E\_() {

if (lookahead == '+') {

match('+');

T();

E\_();

} // E' -> ε (do nothing)

}

void T() {

F();

T\_();

}

void T\_() {

if (lookahead == '\*') {

match('\*');

F();

T\_();

} // T' -> ε (do nothing)

}

void F() {

if (lookahead == '(') {

match('(');

E();

match(')');

} else if (isalnum(lookahead)) {

match(lookahead); // match id

} else {

printf("Syntax error: unexpected character %c\n", lookahead);

exit(1);

}

}

void parse(char\* expr) {

input = expr;

lookahead = \*input++;

E();

if (lookahead == '\0') {

printf("Parsing successful\n");

} else {

printf("Syntax error: unexpected character %c at the end\n", lookahead);

}

}

int main() {

char expr[] = "(id + id) \* id";

parse(expr);

return 0;

}

**Output:**

