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

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

char\* input;

char lookahead;

void E();

void E\_prime();

void T();

void T\_prime();

void F();

void match(char t);

void E() {

T();

E\_prime();

}

void E\_prime() {

if (lookahead == '+') {

match('+');

T();

E\_prime();

}

}

void T() {

F();

T\_prime();

}

void T\_prime() {

if (lookahead == '\*') {

match('\*');

F();

T\_prime();

}

}

void F() {

if (lookahead == '(') {

match('(');

E();

match(')');

} else if (isalpha(lookahead)) {

match(lookahead);

} else {

printf("Syntax error\n");

exit(1);

}

}

void match(char t) {

if (lookahead == t) {

lookahead = \*input++;

} else {

printf("Syntax error\n");

exit(1);

}

}

void parse(char\* expr) {

input = expr;

lookahead = \*input++;

E();

if (lookahead == '\0') {

printf("Parsing successful\n");

} else {

printf("Syntax error\n");

}

}

int main() {

char expr[] = "(a+b)\*c";

parse(expr);

return 0;

}

