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

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <string.h>

char input[100];

int index = 0;

void E();

void E\_prime();

void T();

void T\_prime();

void F();

void match(char c) {

if (input[index] == c) {

index++;

} else {

printf("Error: Expected '%c' but found '%c'\n", c, input[index]);

exit(1);

}

}

void E() {

T();

E\_prime();

}

void E\_prime() {

if (input[index] == '+') {

match('+');

T();

E\_prime();

}

}

void T() {

F();

T\_prime();

}

void T\_prime() {

if (input[index] == '\*') {

match('\*');

F();

T\_prime();

}

}

void F() {

if (input[index] == '(') {

match('(');

E();

match(')');

} else if (isalnum(input[index])) {

index++;

} else {

printf("Error: Invalid expression\n");

exit(1);

}

}

int main() {

printf("Enter an expression: ");

scanf("%s", input);

E();

if (input[index] == '\0') {

printf("Expression is valid!\n");

} else {

printf("Error: Invalid expression\n");

}

return 0;

}

OUTPUT

