10.Write a C program to construct recursive descent parsing.

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

void error();

void E();

void Eprime();

void T();

void Tprime();

void F();

void match(char token);

char input[100];

int pos = 0;

int main() {

printf("Enter an expression: ");

fgets(input, sizeof(input), stdin);

input[strcspn(input, "\n")] = '\0';

E();

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

printf("Parsing successful.\n");

} else {

printf("Parsing failed.\n");

}

return 0;

}

void error() {

printf("Parsing error.\n");

exit(1);

}

void match(char token) {

if (input[pos] == token) {

pos++;

} else {

error();

}

}

void E() {

T();

Eprime();

}

void Eprime() {

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

match('+');

T();

Eprime();

}

}

void T() {

F();

Tprime();

}

void Tprime() {

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

match('\*');

F();

Tprime();

}

}

void F() {

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

match('(');

E();

match(')');

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

pos++;

} else {

error();

}

}

**OUTPUT:**

