```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char *input;
char token[10];

void advance();
void E();
void Eprime();
void T();
void Tprime();
void F();

void advance() {
    while (*input == ' ')
        input++;
    if (*input == '\0') {
        strcpy(token, "$");
        return;
    }
    if (strncmp(input, "id", 2) == 0 && (input[2] == '\0' || input[2] == '+' || input[2] == '*' || input[2] == ')' || input[2] == '(')) {
        strcpy(token, "id");
        input += 2;
    } else if (*input == '+' || *input == '*' || *input == '(' || *input == ')') {
        token[0] = *input;
        token[1] = '\0';
        input++;
    } else {
        printf("Error: invalid symbol near '%c'\n", *input);
        exit(1);
    }
}

void E() {
    T();
    Eprime();
}

void Eprime() {
    if (strcmp(token, "+") == 0) {
        advance();
        T();
        Eprime();
    }
}

void T() {
    F();
    Tprime();
}
```

```
void Tprime() {
    if (strcmp(token, "*") == 0) {
        advance();
        F();
        Tprime();
    }
}

void F() {
    if (strcmp(token, "id") == 0) {
        advance();
    } else if (strcmp(token, "(") == 0) {
        advance();
        E();
        if (strcmp(token, ")") == 0)
            advance();
        else {
            printf("Error: missing closing parenthesis\n");
            exit(1);
        }
    } else {
        printf("Error: invalid token '%s'\n", token);
        exit(1);
    }
}
```

```
int main() {
    char expr[100];
    printf("Enter expression: ");
    if (!fgets(expr, sizeof(expr), stdin))
        return 1;

    size_t len = strlen(expr);
    if (len > 0 && (expr[len - 1] == '\n' || expr[len - 1] == '\r'))
        expr[len - 1] = '\0';

    input = expr;
    advance();
    E();

    if (strcmp(token, "$") == 0)
        printf("Input accepted: Valid expression\n");
    else
        printf("Error: invalid expression near '%s'\n", token);

    return 0;
}
```