```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
char input[100];
int i = 0;
int E();
int EP();
int T();
int TP();
int F();
int main() {
    int choice;
    printf("\nRecursive descent parsing for the following grammar\n");
    printf("E  -> T E'\n");
    printf("E' -> + T E' / ε\n");
    printf("T  -> F T'\n");
    printf("T' -> * F T' / ε\n");
    printf("F  -> (E) / id\n");
    do {
        i = 0;
        printf("\nEnter the string to be checked: ");
        fgets(input, sizeof(input), stdin);
        input[strcspn(input, "\n")] = '\0';
        if (E() && input[i] == '\0')
            printf("\n✅ String is accepted\n");
        else
            printf("\n❌ String is not accepted\n");
        printf("\nDo you want to test another string? (1=Yes / 0=No): ");
        scanf("%d", &choice);
        getchar();
    } while (choice == 1);

    return 0;
}
```

```
int E() {
    if (T()) {
        if (EP()) return 1;
    }
    return 0;
}
int EP() {
    if (input[i] == '+') {
        i++;
        if (T() && EP()) return 1;
        return 0;
    }
    return 1;
}
int T() {
    if (F()) {
        if (TP()) return 1;
    }
    return 0;
}
int TP() {
    if (input[i] == '*') {
        i++;
        if (F() && TP()) return 1;
        return 0;
    }
    return 1;
}
```

```
int F() {
    if (input[i] == '(') {
        i++;
        if (E() && input[i] == ')') {
            i++;
            return 1;
        }
        return 0;
    } else if (isalpha(input[i])) {
        i++;
        return 1;
    }
    return 0;
}
```

```
Recursive descent parsing for the following grammar
E  -> T E'
E' -> + T E' / ε
T  -> F T'
T' -> * F T' / ε
F  -> (E) / id

Enter the string to be checked: (a+b)*c

✅ String is accepted

Do you want to test another string? (1=Yes / 0=No): 1

Enter the string to be checked: a/c+d

❌ String is not accepted

Do you want to test another string? (1=Yes / 0=No): 0


=== Code Execution Successful ===
```