

# 表达式检查

输入若干个C语言的表达式，判断表达式是否合法以及是哪个类型的表达式。考虑的表达式类型包括：算术表达式、关系表达式和逻辑表达式。表达式的合法性检查包括：标识符是否合法及单/双目运算符的合法性（如双目运算符要有两个操作数）。

## 表达式类别的定义

- 算术表达式、关系表达式和逻辑表达式。（操作数可以是标识符或者数字，数字可以是整数或浮点数，可以带上单目运算符）
- 运算符定义以手册为准：[英文界面](#)或[中文界面](#)
- 涉及到的双目运算符如下
  - 算术运算符 + - \* / % | & ^ >> <<
  - 关系运算符 < > <= >= == !=
  - 逻辑运算符 && ||
- 表达式中不会出现括号
- 在一个表达式中，最后计算的运算符类别（按书P245运算符优先级）决定了表达式类别
- 涉及到的单目运算符如下
  - 只允许出现在数字和标识符左侧(最多连续出现一个)的算术运算符 + -
  - 只允许出现在标识符左侧的 \* & ~ !(可以嵌套使用如 !!!!~\*&a)
    - 其中\*和&是成员访问运算符，他们不影响表达式的类别
    - ~是算数运算符，!是逻辑运算符

需要注意的是 如果输入的表达式是 `a &a *a *&a`，需要将它识别为非法表达式

## 非法表达式

1. 单目运算符出现的位置和要求不符 `a != b` (! 与 = 之间有空格)
2. 表达式缺少操作数
3. 表达式中出现非法字符(只能出现上述的单目和双目运算符以及数字和标识符，标识符只含大小写字母)

注意 000012是合法的，12是合法的， 00.12是合法的， 1. 是合法的， .000是合法的， . 是不合法的(只有一个小数点是不可以的)

## 合法表达式

- `-a + b - c * !&d - 0.3`

算术表达式 包含的双目运算符分别是：+ - \* - ，操作数分别是：`-a b c !&d 0.3`

- `d <= e + b - c * *x`

关系表达式 包含的双目运算符分别是<= + - \* 操作数分别是`d e b c *x`

- `a || b | c && d ^ e || f & 0 | 1`

逻辑表达式 包含的双目运算符分别是|| | && ^ || & | 操作数分别是`a b c d e f 0 1`

## 输入形式

一行字符串，为了简化题目难度，其中双目运算符两侧有空格且空格仅在双目运算符两侧出现

## 输出形式

若表达式非法，则输出内容为"Invalid!"

若表达式合法，则输出表达式类型"arithmetic expression"(算术表达式)/"logic expression"(逻辑表达式)/"comparison expression"(关系表达式)

---

## TIPS

这次实验难度较高，请尽量早开始动笔

为了简化实验，我们**不要求**你通过全部的测试样例：在下方的20个测试样例中，通过10个测试样例即可获得全部分数。但是我们鼓励想训练代码能力的同学挑战通过全部测试样例

- 你可以使用<string.h>头文件的strlen()、strtok()库函数：

- `strlen( const char *str );` [strlen, strnlen s - cppreference.com](http://strlen.strnlen.s-cppreference.com)

◦ `char *strtok( char *str, const char *delim );` [strtok, strtok\\_s - cppreference.com](http://strtok.strtok_s-cppreference.com)

- 你可以先根据空格将输入的字符串划分成若干个子串：这些子串或者是双目运算符，或者是由单目运算符和操作数结合形成的子串
- 之后分别判断他们是否合法（双目运算符是否合法，双目运算符的两个操作数是否缺失，操作数是否合法）
- 你需要将子串的表达式类型分析并记录下来，然后根据双目运算符分析表达式类型

## 测试样例

标记为*bonus*的部分较为困难，可以选择通过其它测试

//算术表达式

```
-a + b - c * !&d - 0.3;
```

//关系表达式

```
d <= e + b - c * *x;
```

//逻辑表达式

```
a || b | c && d ^ e || f & 0 | 1;
```

//非法 使用了保留字符int

```
a || b | int && d ^ e || f & 0 | 1; //bonus
```

//非法 逻辑运算符两操作数必须为整型

```
a || b | c && d ^ e || f & 0 | 1.3;
```

//逻辑表达式

```
a || b | c && d ^ e || f & 0LL | 1; //bonus
```

//算术表达式

```
~!&a;
```

//非法 出现--

```
--!*&a;
```

//非法 无效双目运算符@

```
a @ b ^ c;
```

//非法 双目运算符操作数缺失（这里< =是两个双目运算符）

```
a < = b;
```

//非法 不是三种表达式之一

```
*&a;
```

//非法 不是三种表达式之一

```
a = b + c;
```

//非法 双目运算符操作数缺失

```
&&a = c | d;
```

//关系表达式

```
aishodbioascyhasioydasudjwioeqrohiwqh <=
yuasidgyuaistdyui23uiwqyasuiduasidhasdasd;
```

//关系表达式

```
$a < op; //bonus
```

//关系表达式

```
__ptr < 0; //bonus
```

//逻辑表达式

```
- __ || + __; //bonus
```

//关系表达式

```
~asd <= 0;
```

//逻辑表达式

```
!~a;
```

//算术表达式

```
~!a;
```

## 思考与展望

- 可以考虑如果输入字符串在双目运算符两边不添加空格该如何处理
- 如果有括号该如何处理