

# CPP Project 1—— *Calculator*

---

王霄骏 12012018

## 一. 启动计算器

程序使用了Make方法进行编译汇编和连接

```
make //可执行文件生成
```

文件:

- calculator.c
- functions.h
- isInteger.c
- isNumber.c
- Makefile

Makefile文件:

```
calculator: calculator.o isInteger.o isNumber.o
    gcc -o calculator calculator.o isInteger.o isNumber.o
calculator.o: calculator.c
    gcc -c calculator.c
isInteger.o: isInteger.c
    gcc -c isInteger.c
isNumber.o: isNumber.c
    gcc -c isNumber.c
clean:
    rm calculator calculator.o isInteger.o isNumber.o
```

后续按照 **num1 符号 num2** 的格式进行输入即可

```
./calculator 1 + 1 // 示例
```

---

## 二. 基础功能

### 1. 加法(+)

```
./calculator 145 + 347
# 输出应为:
145 + 347 = 492
```

### 2. 减法(-)

```
./calculator 87 - 23
# 输出应为:
87 - 23 = 64

./calculator 45 - 123
# 输出应为:
45 - 123 = -78
```

### 3. 乘法(x)

**注意：计算器使用的小写字母x作为乘法的符号**

```
./calculator 123 x 456
# 输出应为:
123 x 456 = 56088
```

### 4. 除法(/)

```
./calculator 763 / 123
# 输出应为:
763 / 123 = 6.20325203
```

---

## 三. 异常处理

### 1. 输入的不是数字:

```
./calculator a x r
# 输出 (两个num分别输出ERROR)
ERROR: Input num1 is not a number!!
ERROR: Input num2 is not a number!!
```

### 2. 输入的符号异常

```
./calculator 12 * 34
# 输出
ERROR: The input operator is illegal
```

### 3. 输入0作为除法的除数

```
./calculator 3.14 / 0
# 输出
ERROR: The divisor cannot be 0!!
```

---

## 四. 特殊情形下的处理

### 1. 大数字的乘法

```
./calculator 987654321 x 123456789
# 输出
987654321 x 123456789 = 121932631112635269
```

---

## 五. 总结

整体的代码逻辑是:

- 将所有的输入先用long double进行存储以及运算，最后输出的时候在进行判断
- 如果为整数则把小数点和0去掉输出，否则就按小数输出。

我声明了两个方法分别来判断是否为整数以及是否为数字

```
int isInteger(char *n);
int isNumber(char *n);
```

计算器总的思路很简单，但是想要同时兼顾大数字，整数，小数的输入输出，异常处理，需要考虑的东西还是很多的，尤其对于刚刚学习c语言的我来说很多的方法很多的库都是在网上现查的，学习怎么使用也花费了挺多的时间。

总的来说算是一个入门的好项目，能够熟悉c的语法和一些常见库的调用，对于数字的处理也能够使对于c中一些奇怪的特性有更好的理解。