

编码规范

1.排版风格

<规则 1>

程序块采用缩进风格编写，按照软件正常缩进。

<规则2>

在两个以上的关键字、变量、常量进行对等操作时，它们之间的操作符之前、之后或者前后要加空格；进行非对等操作时，如果是关系密切的立即操作符(如->)，后不应加空格，关键字两侧加空格。

- 例如
- if () ..., 不要写成if() ...
- 逗号、分号只在后面加空格
printf("%d %d %d", a, b, c);
- "=", "+=", "+", "%", "&&", "&", "<<", "^"等操作符的不加空格

<规则3>

每一个模块都要注释是干什么用的，且模块与模块之间要空行

例如：

```
.....  
}  
  
void bracket(int m[],char n[],int t) //添加括号  
{  
    .....
```

<规则4>

if、while、for、case、等语句自占一行，并且循环、判断语句的程序块部分用花括号括起来，即使只有一条语句。

例如应如下书写：

```
if( bCondition == TRUE )  
{  
    bFlag = YES;  
}
```

2. 可理解性

2.1 注释

何时需要注释：

- 如果变量的名字不能完全说明其用途，应该使用注释加以说明。
- 如果为了提高性能而使某些代码变得难懂，应该使用注释加以说明。
- 对于一个比较长的程序段落，应该加注释予以说明。如果设计文档中有流程图，则程序中对应的位置应该加注释予以说明。
- 如果程序中使用了某个复杂的算法，建议注明其出处。
- 如果在调试中发现某段落容易出现错误，应该注明。

<规则1>

程序在必要的地方必须有注释，注释要准确、易懂、简洁，在设变量的时候要第一次说明此变量的作用，若代码很长，要偶尔重复提示此变量的作用 例如：

```
Boolean w: //是否第一次做加减法
.....
if ( w == true ) //如果是第一次做加减法
```

<规则2>

注释一律放在右面

2.2 命名

<规则1>

函数名用本函数的作用的关键词 例如：

```
void equation( int t) //编写方程
void bracket(int m[],char n[],int t) //添加括号
```

<规则2>

变量的命名尽量简单，去英文单词的前几个字母，或取大写字母

1.3 可维护性

<规则1>

避免使用 goto 语句。

<规则2>

在逻辑表达式中使用明确的逻辑判断。

```
if ( w == true )    //而不能if ( w )
```

1.4. 程序正确性、效率

<规则1>

变量必须初始化

<规则2>

定义公共指针的同时对其初始化。

<规则4>

防止内存操作越界。

A.数组越界。

B.指针操作越界。

1.5 接口

<规则1>

去掉没有必要的公共变量，编程时应尽量少用公共变量。

<规则2>

返回值为指针的函数，不可将局部变量的地址作为返回值。

<规则5>

尽量不设计多参数函数，将不使用的参数从接口中去掉，降低接口复杂度。

参考：<http://www.cnblogs.com/samuelwnb/p/3868971.html>

<http://home.ustc.edu.cn/~danewang/c/CodingStandards.html>

