

# 实训要求及说明

C 语言程序设计的实训是学生重要的实习环节。上机实训不仅要求学生掌握 C 语言程序设计的基本知识，更重要的是培养学生掌握程序设计开发的基本素质、思维方法和技能，为学生综合素质的培养打下坚实基础。实训的形式为教师制定出难易适中的设计题目，学生通过分组讨论进行设计，实训重点放在程序设计开发的全过程上。

## 一、总要求

根据所选题目首先进行年算法分析，根据实现的功能进行数据分析，设计相应的数据结构，同时为贯彻“程序设计=算法+数据结构”的思想，要求学生在课下完成流程图的设计，然后根据流程图写出实训代码。

## 二、代码书写要求

为保证开发团队的协作和后期修改能有效进行，使学生遵照统一的排版风格、注释标准、命名规则及编码诸原则进行 C 语言编程，特提出以下规范要求。

### （一）缩进

1、程序块要采用缩进风格编写，缩进以 4 个空格（Space）键为单位，不使用"TAB 键"。

2、函数体、结构体、循环体以及分支结构中的语句行都须采用缩进风格。

3、所有的 if、while、for、do 结构中的语句即使只有一行也须用括号括起来。

4、if、while、for、do 语句单独占一行，左、右花括号也各占一行且不缩进。

例：

```
if(superHero == theTick)
{
    printf("Spoon!");
}
```

### （二）间隔

所有的标识符都必须被空白字符包围。

例如：

```
int theTick = 5;
if (theTick == 5)
```

### （三）空行

代码段之间须以一个空行进行间隔。

#### （四）空格的位置

- 1、在一个关键字和左括号“(”之间。注意：不要在函数名和左括号之间加空格；
- 2 在参数列表的每个逗号“,”之后；
- 3、二元操作符前后。注意：一元操作符前后都不加空格。例如：`int a = 10; a = a + 1; a++;`
- 4、for 语句的每个表达式之间。例如：`for (int i = 0; i < 20; i++)...`

#### （五）代码行

- 1、不允许把多个短语写在一行中，即一行只写一条语句；
- 2、长语句、参数、表达式（超过 80 个 ASCII 字符）应分行书写，操作符放在新行之首，划分出的新行要进行适当的缩进，使排版整齐，语句可读。

示例：

```
#001 perm_count_msg.head.len = NO7_TO_STAT_PERM_COUNT_LEN
#002   + STAT_SIZE_PER_FRAM ;
#003 act_task_table[STAT_TASK_CHECK_NUMBER + index].occupied
#004   = stat_poi[index].occupied;
```

#### （六）命名规则

所有的标识符只能用字母（A-Z 或 a-z）和数字（0-9）。不得包含货币符号或其它非 ASCII 字符。

- 1、变量的命名应尽可能采用有意义的名词，力求可顾名思义。
- 2、可以采用一个单词或多个单词的缩写作为名字，缩写单词的每个字母都要大写。
- 3、对于难以使用英文的情况，可以参考相关行业标准，比如使用国标。
- 4、采用约定俗成的习惯用法。

常见的习惯用法：

- （1）循环变量：i、j、k、m、n
- （2）长度：length
- （3）数量：count
- （4）位置：pos 或 position
- （5）下标或索引：i 或 index
- （6）设置/获取：set/get
- （7）大小：size

#### （七）注释

- 1、为确保程序易读性，注释语句不得小于全部代码行的 30%。
- 2、说明性文件（如头文件.h 文件、.inc 文件、.def 文件、编译说明文件.cfg 等）头部应进行注释，注释必须列出：版权说明、版本号、生成日期、作者、内容、功能、与其它文件的关系、修改日志等，头文件的注释中还应包含函数功能简要说明。
- 3、源文件头部应进行注释，列出：版权说明、版本号、生成日期、作者、模块目的/功能、主要函数及其功能、修改日志等。
- 4、函数头部应进行注释，列出：函数的目的/功能、输入参数、输出参数、

返回值、调用关系（函数、表）等。

5、边写代码边注释，修改代码同时修改相应的注释，以保证注释与代码的一致性。不再有用的注释要删除。

6、注释的内容要清楚、明了，含义准确，防止注释二义性。

7、避免在注释中使用缩写，特别是非常用缩写。

8、注释应与其描述的代码相近，对代码的注释应放在其上方或右方（对单条语句的注释）相邻位置，不可放在下面，如放于上方则需与其上面的代码用空行隔开。

9、对于所有有物理含义的变量、常量，如果其命名不是充分自注释的，在声明时都必须加以注释，说明其物理含义。变量、常量、宏的注释应放在其上方相邻位置或右方。

10、数据结构声明(包括数组、结构、类、枚举等)，如果其命名不是充分自注释的，必须加以注释。对结构中的每个域的注释放在此域的右方。

11、全局变量要有较详细的注释，包括对其功能、取值范围、哪些函数或过程存取它以及存取时注意事项等的说明。

12、注释与所描述内容进行同样的缩排。

13、将注释与其上面的代码用空行隔开。

14、对变量的定义和分支语句（条件分支、循环语句等）必须编写注释。

15、对于 switch 语句下的 case 语句，如果因为特殊情况需要处理完一个 case 后进入下一个 case 处理，必须在该 case 语句处理完、下一个 case 语句前加上明确的注释。

### 三、编码原则

1、注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级。

2、避免使用不易理解的数字，用有意义的标识来替代。涉及物理状态或者含有物理意义的常量，不应直接使用数字，必须用有意义的枚举或宏来代替。

3、去掉没必要的公共变量。

4、仔细定义并明确公共变量的含义、作用、取值范围及公共变量间的关系。

5、明确公共变量与操作此公共变量的函数或过程的关系，如访问、修改及创建等。

6、当向公共变量传递数据时，要十分小心，防止赋予不合理的值或越界等现象发生。

7、防止局部变量与公共变量同名。

8、严禁使用未经初始化的变量。

9、对所调用函数的错误返回码要仔细、全面地处理。

10、明确函数功能，精确（而不是近似）地实现函数设计。