计算机学习报告

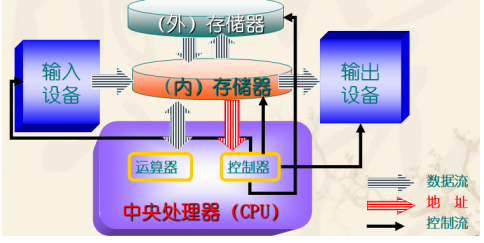
20049200452黄哲

1. 计算机导论

**一、计算机工作原理：“存储程序”+“程序控制**”

1、以二进制形式表示数据和指令

2、将程序存入存储器中，由控制器自动读取并执行

3、外部存储器存储的程序和所需数据 =⇒ 计算机内存 =⇒ 在程序控制下由 CPU 周而复始地取出指令、分析指令、执行指令 =⇒ 操作完成。

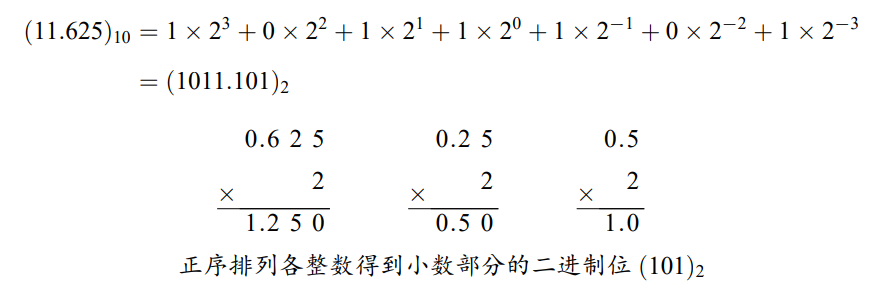
（二）程序设计

**一、十进制（D）、二进制（B）和十六进制（H，在c语言中为0x）**

1、(D)10 = Dn−1 × 10n−1 + Dn−2 × 10n−2 + · · · + D1 × 101 + D0 × 100 + D−1 × 10−1 + D−2 × 10−2 + · · · + D−m+1 × 10−m+1 + D−m × 10−m

(B)2 = Bn−1 × 2 n−1 + Bn−2 × 2 n−2 + · · · + B1 × 2 1 + B0 × 2 0 + B−1 × 2 −1 + B−2 × 2 −2 + · · · + B−m+1 × 2 −m+1 + B−m × 2 −m

(H)16 = Hn−1 × 16n−1 + Hn−2 × 16n−2 + · · · + H1 × 161 + H0 × 160 + H−1 × 16−1 + H−2 × 16−2 + · · · + 16−m+1 × 16−m+1 + H−m × 16−m

小数转二进制是乘二取整

eg：

注意：用二进制表示小数位存在精度问题

2、进制对照表：

3、数值在计算机中的表示

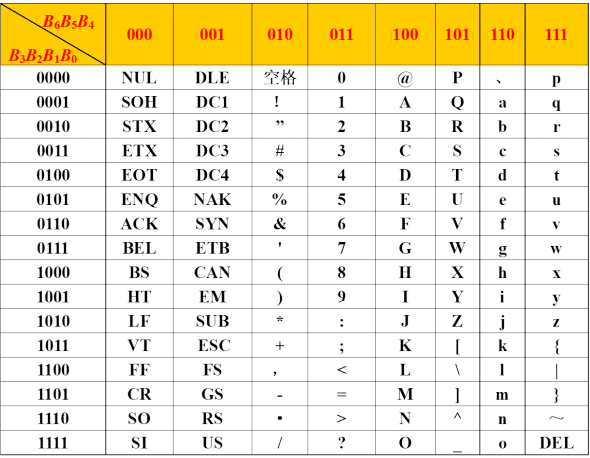
* 原码：正数的符号为 0，负数的符号为 1，其它位按一般的方法表示数的绝对值。

Eg：x = (+103)10 [x]原 = (0110 0111)2 ；x = (−103)10 [x]原 = (1110 0111)2

* 反码：符号位不变，正数鱼原码相同，负数其余位与原码相反（0→1，1→0）
* 补码：正数就是原码，负数为反码+1（好处：将减法计算全部转换为加法）

Eg：

4、ASCⅡ编码B6B5B4B3B2B1B0



数字=编码值-‘0’

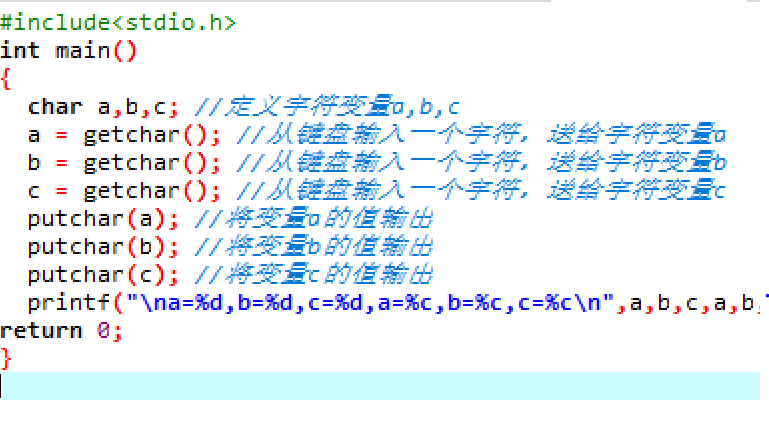
E

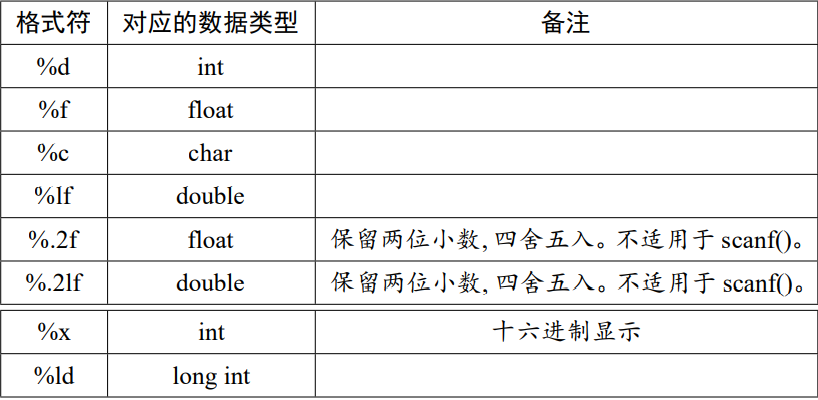
‘小写字母’=’大写字母’+32

**二、课堂程序讲解**

1、定义数据类型以及对应格式描述符

int：整数类型；float：单精度浮点数；double：双精度浮点数；char：字符类型；getchar：接收字符；putchar：输出接收的字符

Eg：

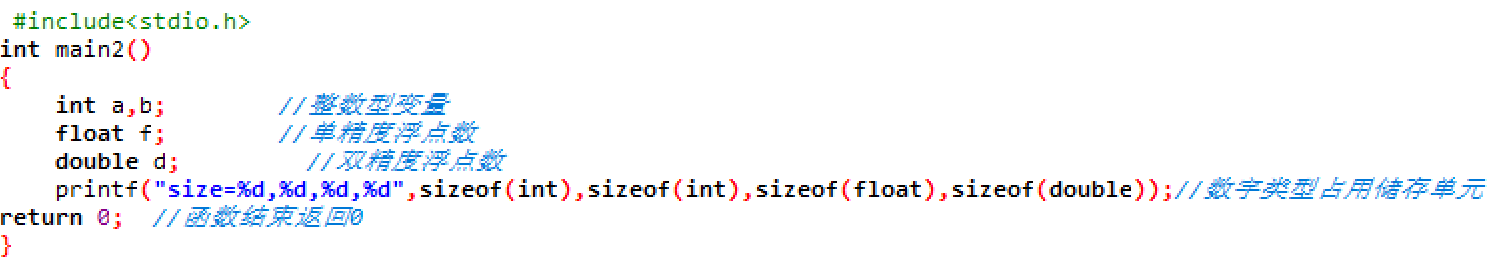
数据类型及其对应格式描述符

输入与输出语句：输入：scanf(“d%”,&a) 注：&符号不能掉！！

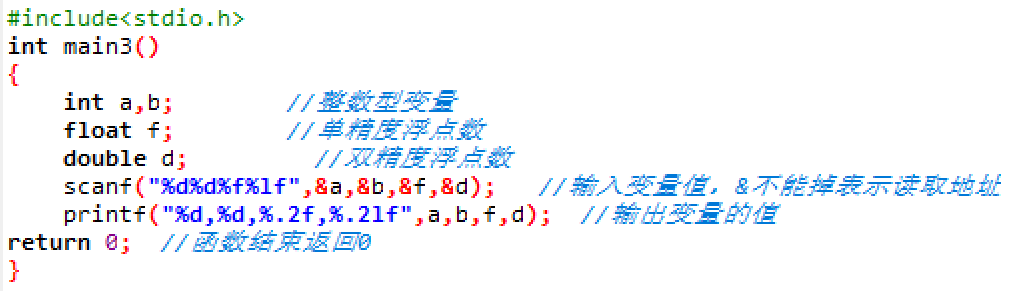
输出：printf(“d%”,a)

示例程序设计：

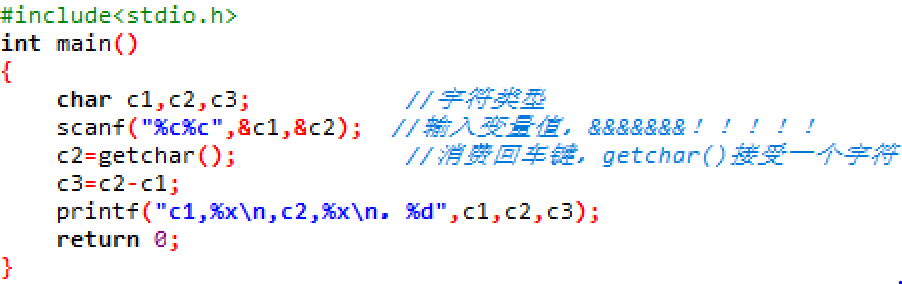
1. 定义数据类型：



1. 输入变量：scanf(“% 变量格式符”, & 变量名); 输出变量：printf(“原样输出, % 格式符”, 对应变量值);



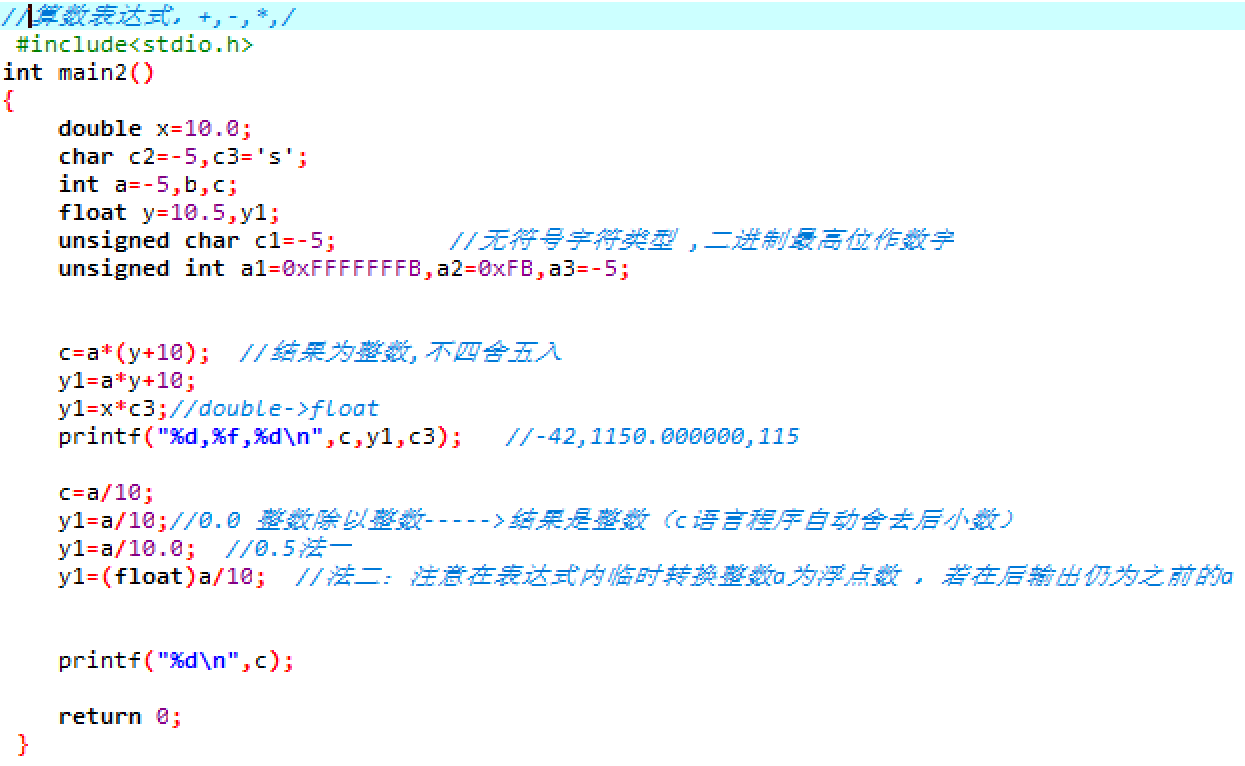
1. 字符类型：



2、

无符号字符类型（unsigned：二进制最高位不作符号位，而当作是数字）

算数表达式（+，-，\*，/，%取余数） 注：ab在c语言中不存在，应该使用数学函数#include<math.h>pow(a,b)

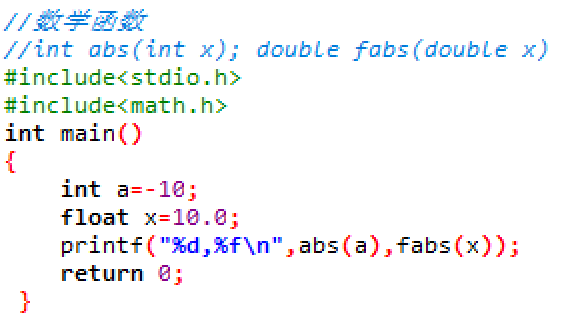


此处整数/整数为整数，注意将整数转换为浮点数

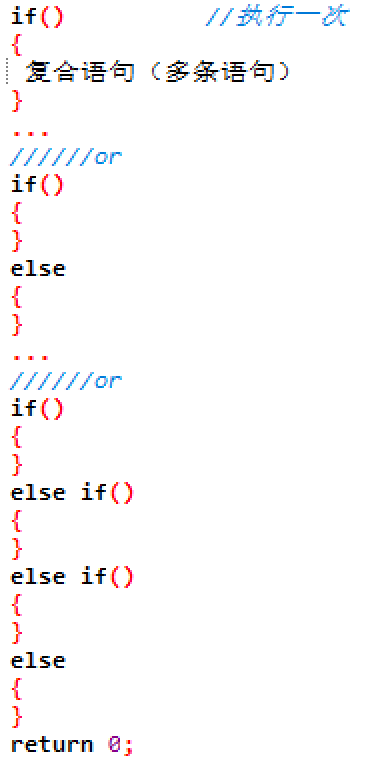
i++与++i的区别：i++先赋值再加一 eg：i=1；a=i++ 🡺 i=2,a=1

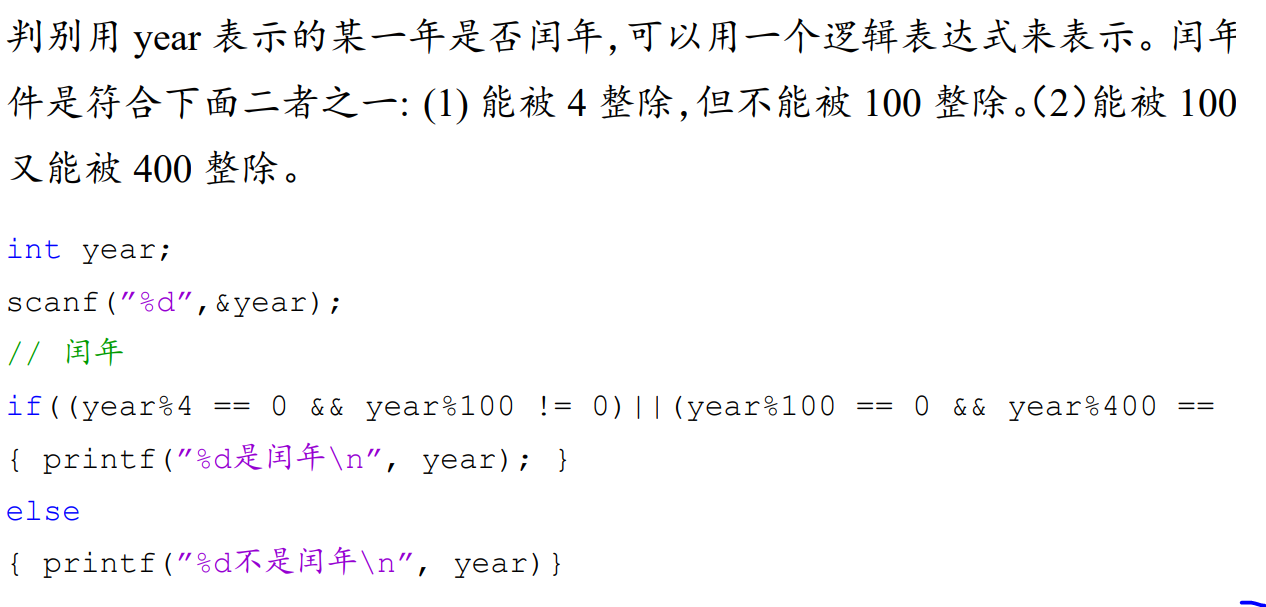
++i先加一再赋值 eg：i=1; a=++i 🡺 i=2,a=2

数学函数（abs()，fabs()）:

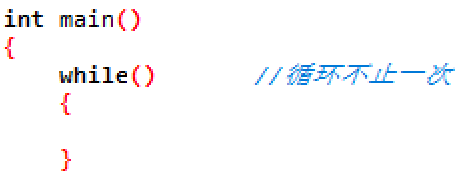


3、条件语句以及循环结构

基本结构1）if条件结构 eg:

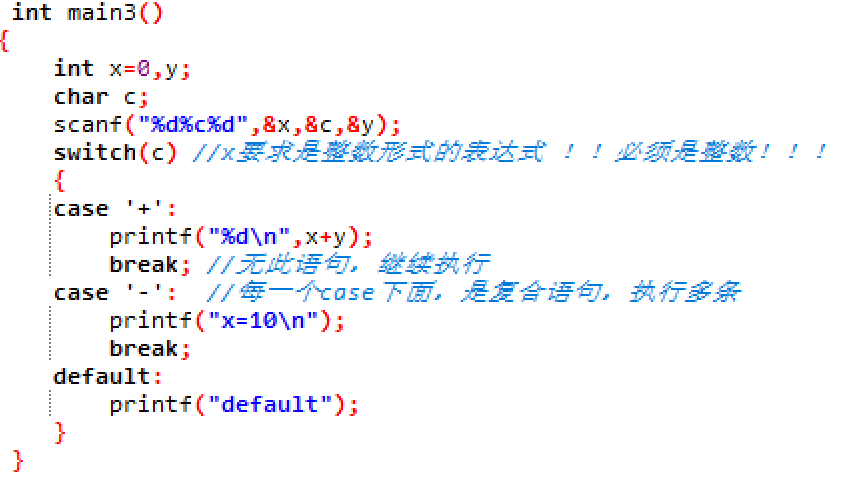


2）while结构

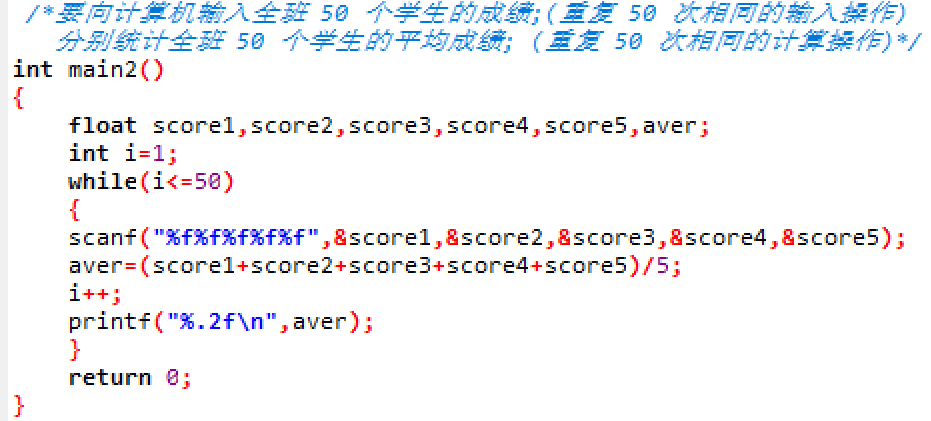


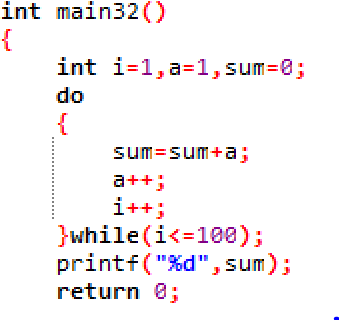
3)switch结构（对于多种情况使用）

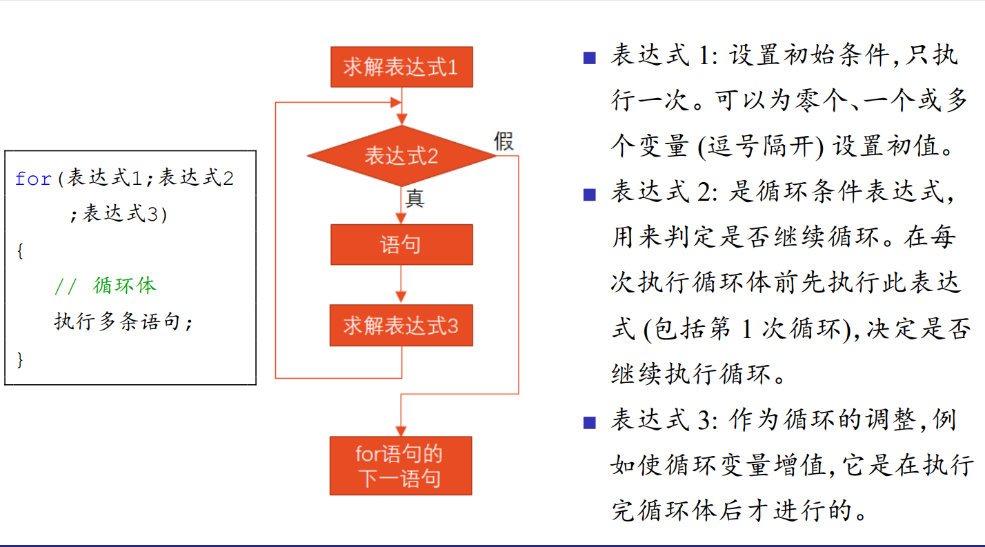
注：每种case后要加break终止循环



4)循环语句

1. While结构：

 b）do{}while（）；结构：



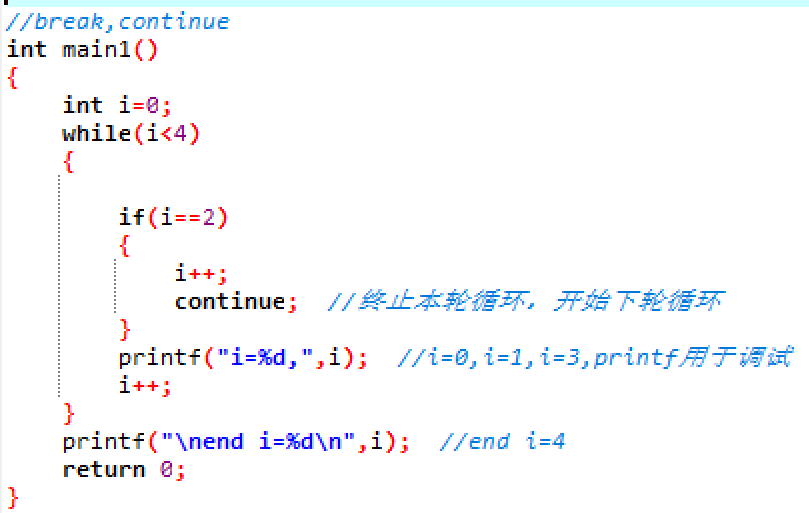
c）for（;;){}结构：

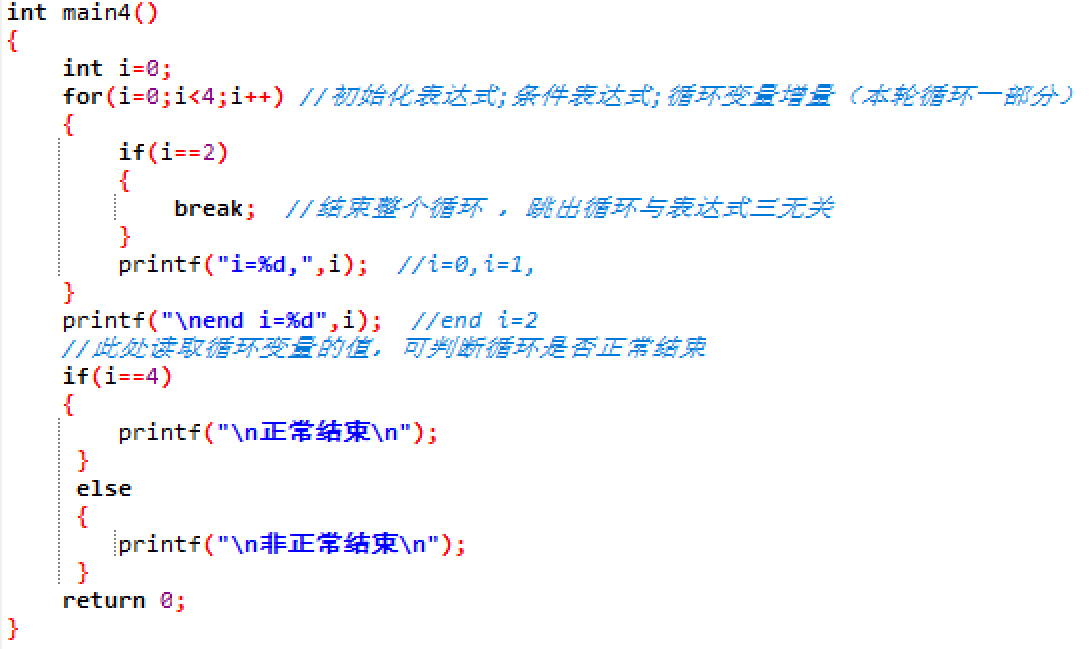
知识点：

1. 定义常数：#define pi 3.1415926
2. 计算机语言与数学表达式不同：a）If（）{}中不为0则视为1表示真，执行花括号中内容，若为0，则视为假不执行，所以，例如，在叙述a=0时，应该描述为a==0

逻辑运算符真值表：

1. 不等式表示不相同，例如，0<=a<=10，在c语言中存在语言错误，正确表达应该为0<=a&&a<=10，运用&&逻辑字符，同时在表示两条件满足其一即可执行的语句时，即为“或”的关系，应使用逻辑字符||
2. Continue与break的区别：continue 语句只结束本次循环，而非终止整个循环。break 语句结束整个 循环，不再判断执行循环的条件是否成立。 continue即在满足条件的情况下，不执行表达式三，继续下一轮循环，break即为终止循环，跳出该循环体系，具体情况如下：





1. Switch语句中每个case后一定要加break，否则不会结束，而且break只能出现在循环语句和switch语句中，不能单独使用。

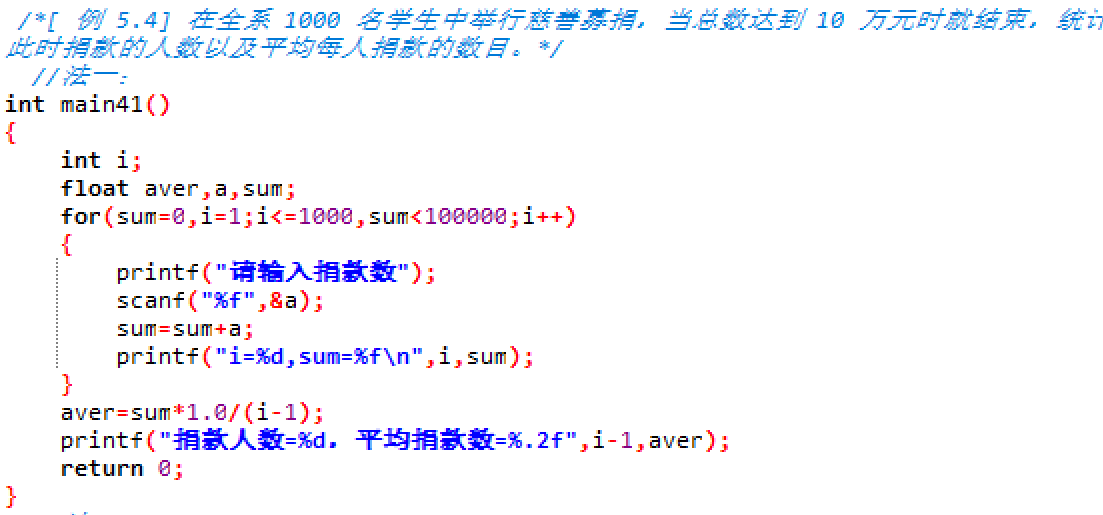
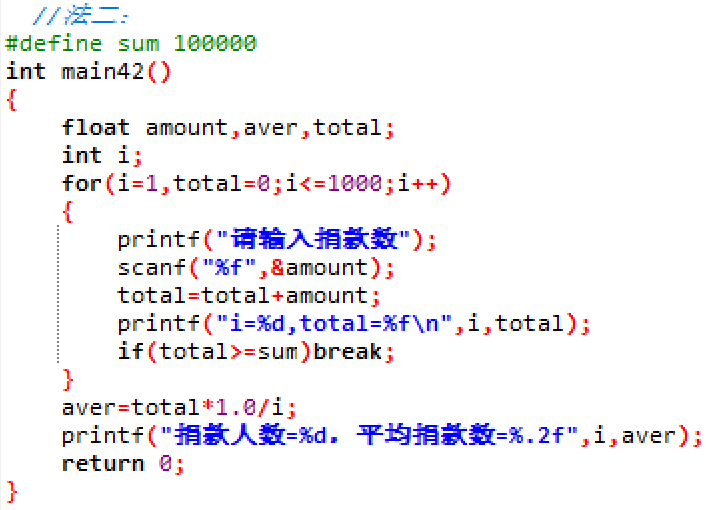
注意事项：1）while( ){ }; do { } while( ); for(;;){ } 执行顺序；

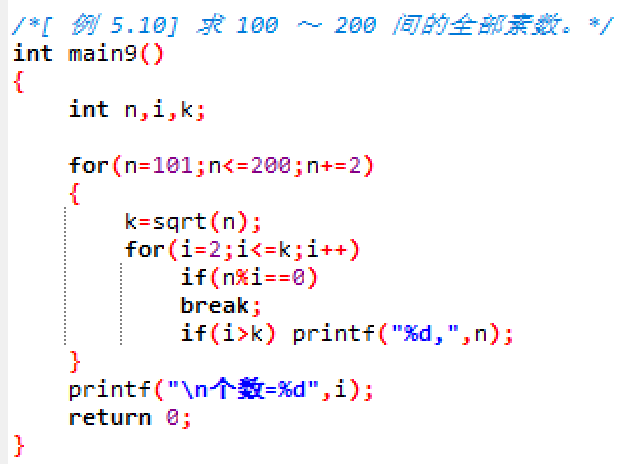
2）循环变量的开始和结束条件；

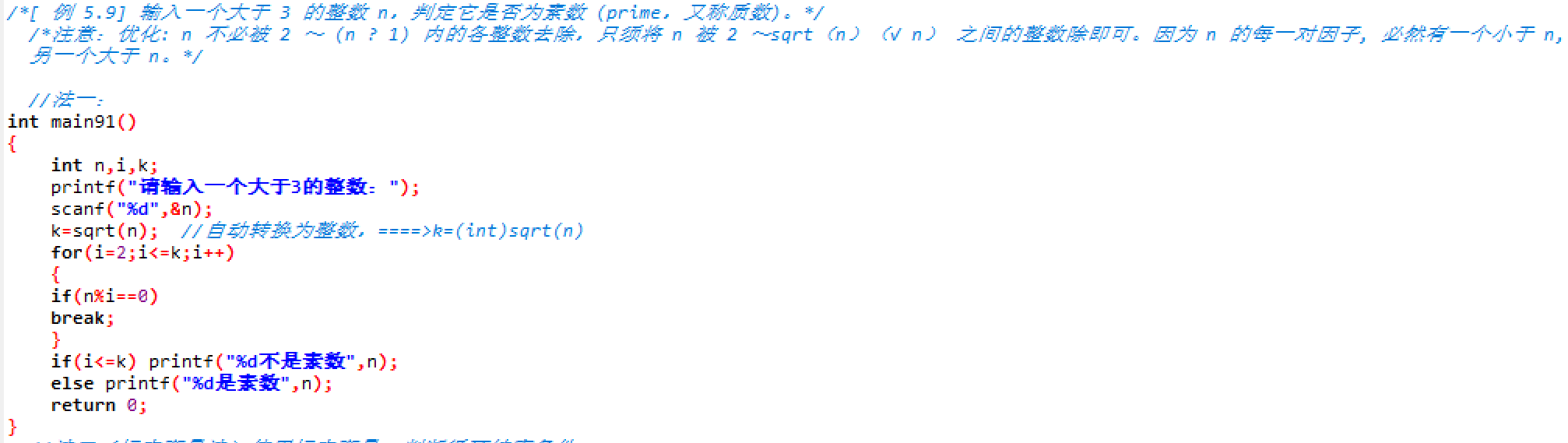
3）循环体是复合语句时，必须用 { } 扩起来；

4）必要时，用 break 结束整个循环，用 continue 结束本次循环；

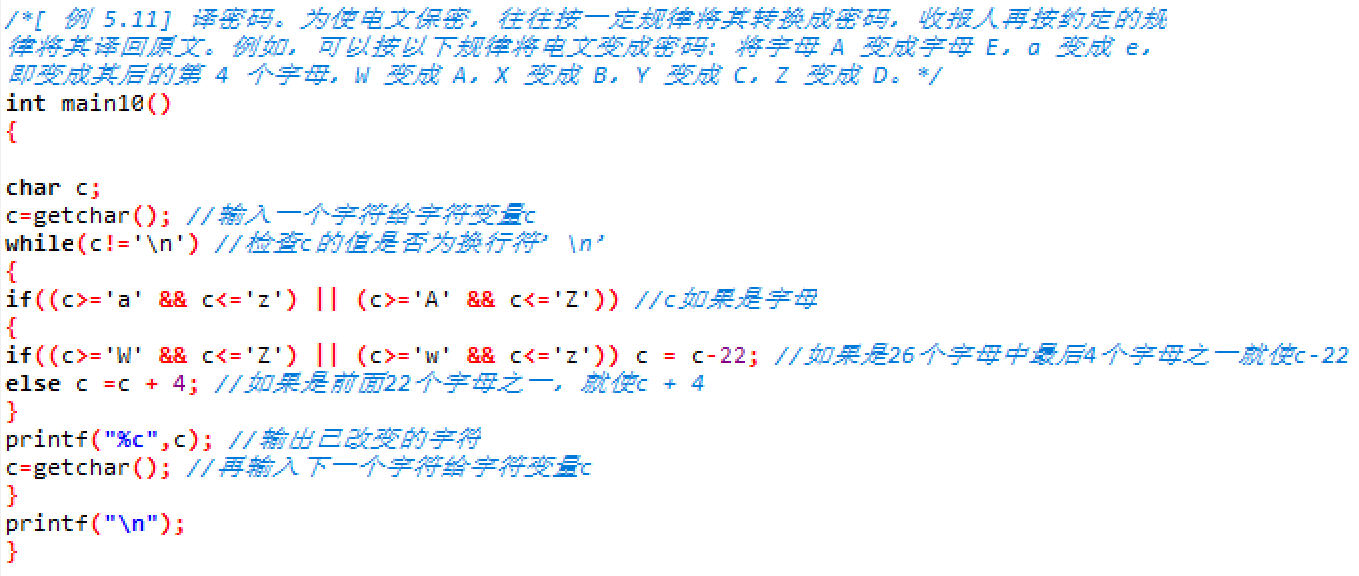
5）关键是找出循环规律，必要时设计流程图，指导代码实现。

典型例题：









3、定义函数

基本格式：函数类型 函数名称（参数类型1，参数类型2，……）

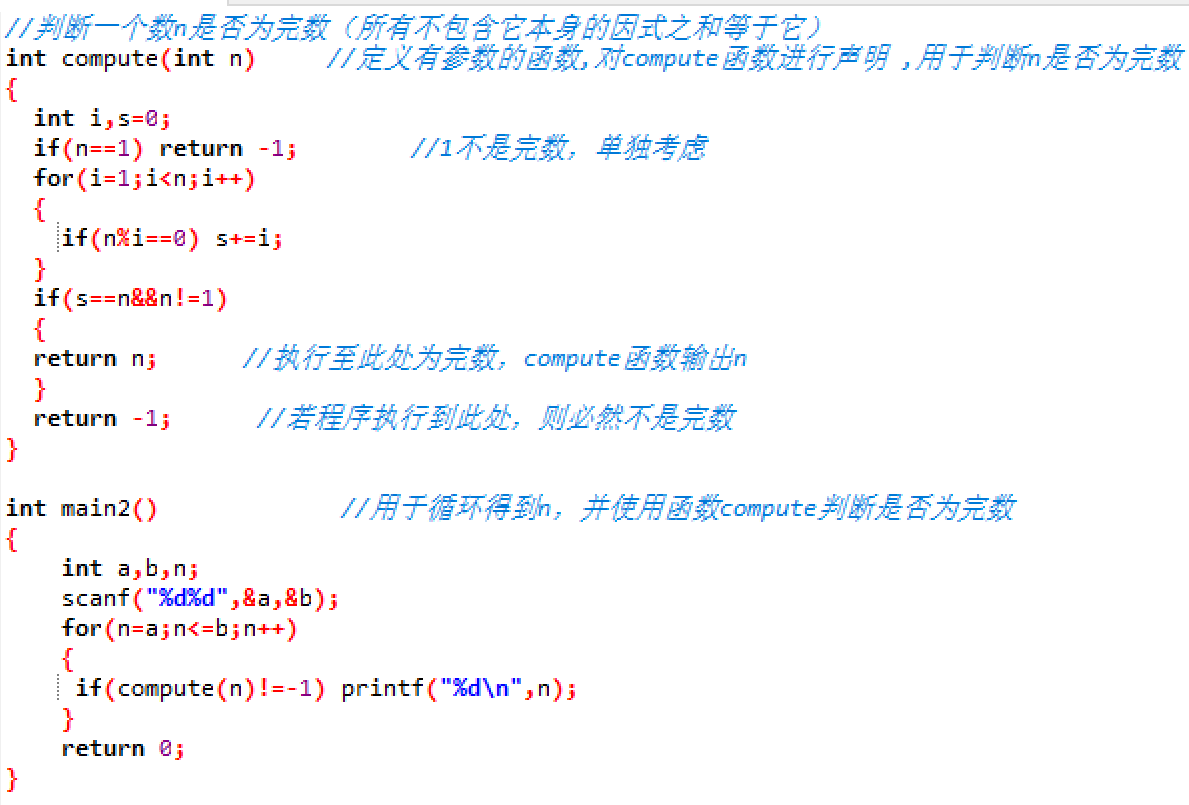
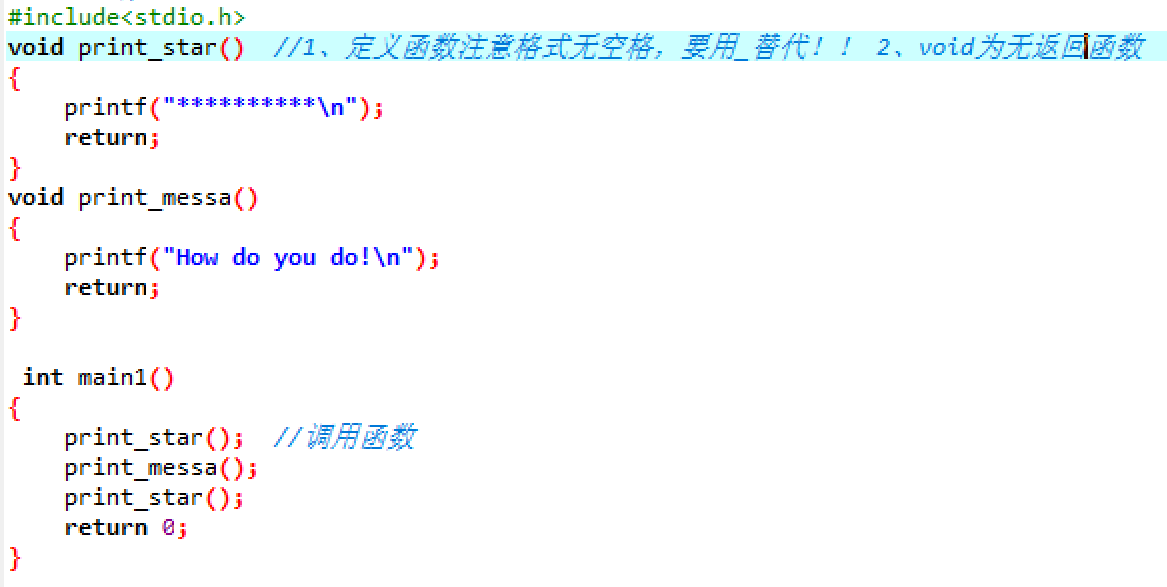
{ 函数主体;}

例如：定义整数型函数 int sum(int x,int y) {………return 0}

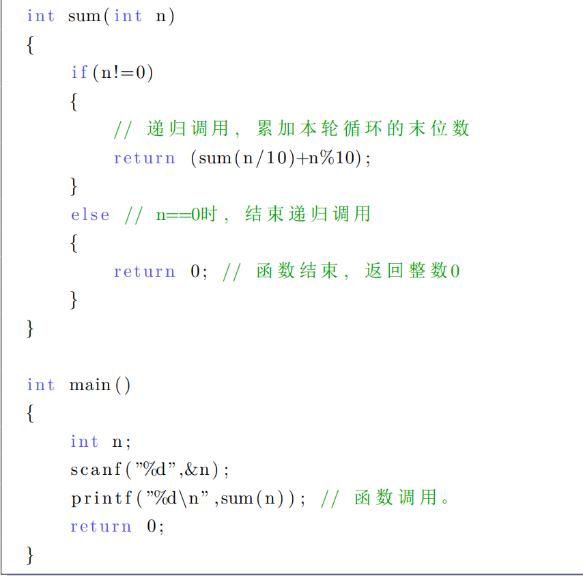
定义无返回型函数 void fun1（int a,int b）{…………return}

例题1：定义无参数函数

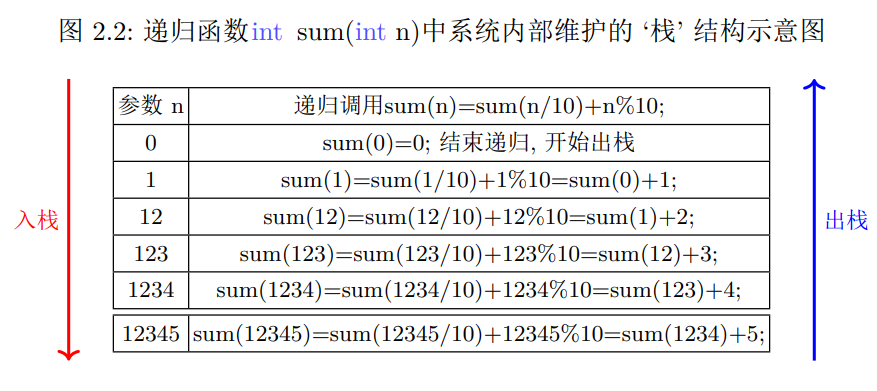
例题2：定义有参数函数，完数的计算



例题3：定义递归函数，在调用的过程中直接或间接使用该函数本身

Eg：求各位数字之和(应用递归函数)

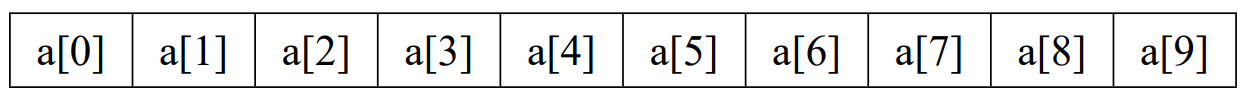
调用递归函数时，系统内部会自动维护一个称为“栈”的存储数据的空间



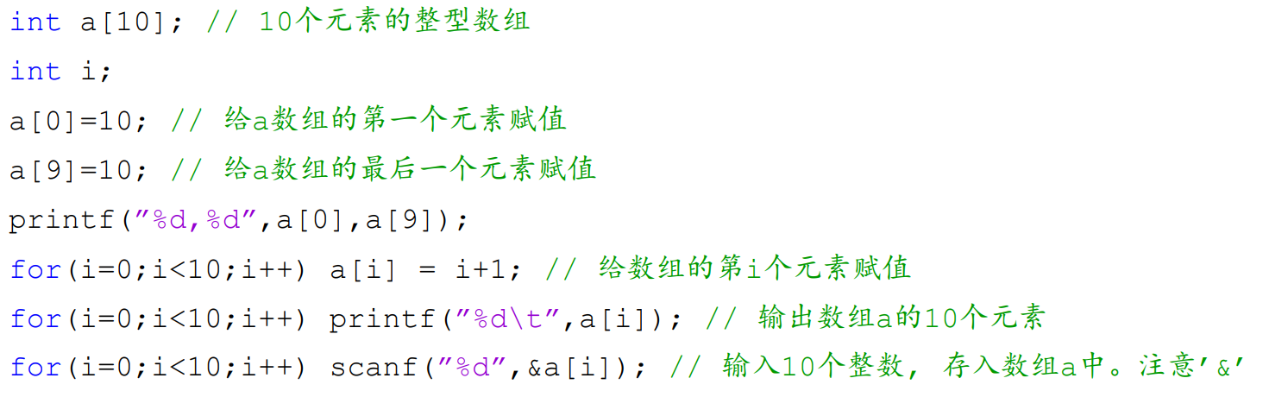
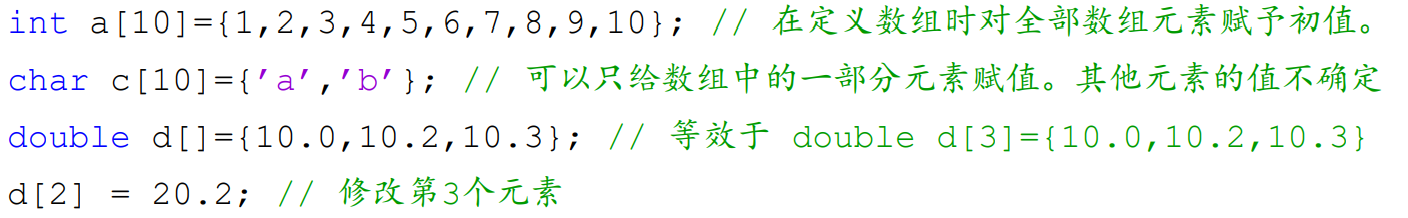
4、数组

定义数组以及数组的引用

定义一维数组: 元素类型 数组名 [常量表达式 (表示元素个数—数组的长度)]

注意：例如定义数组int score[10] 数组score[i]，i为数组下标，从0开始而数组长度只有10，则i只能取到9

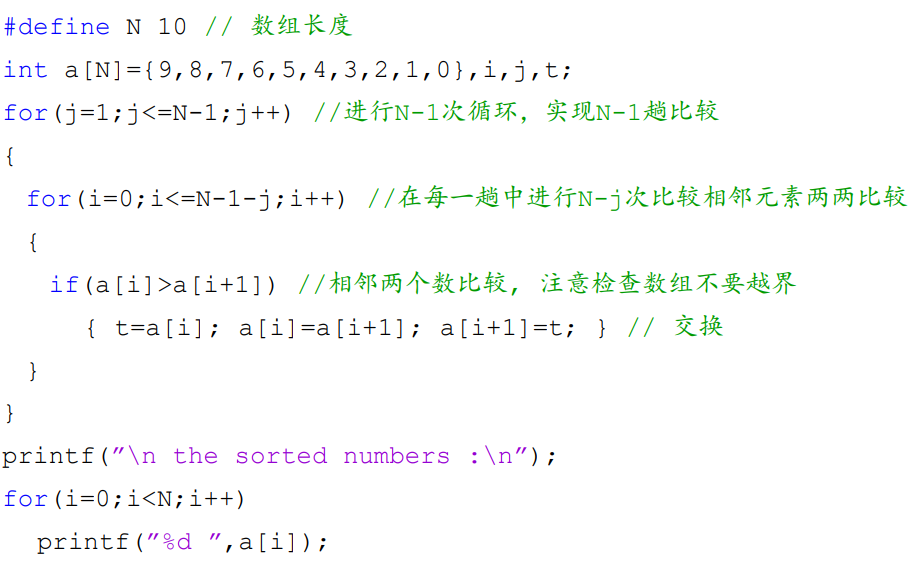
引用数组：

1. 如果数组中数字具有一定的规律，则用for循环给数组定义数字
2. 如果没有规律，则用初始化为数组定义

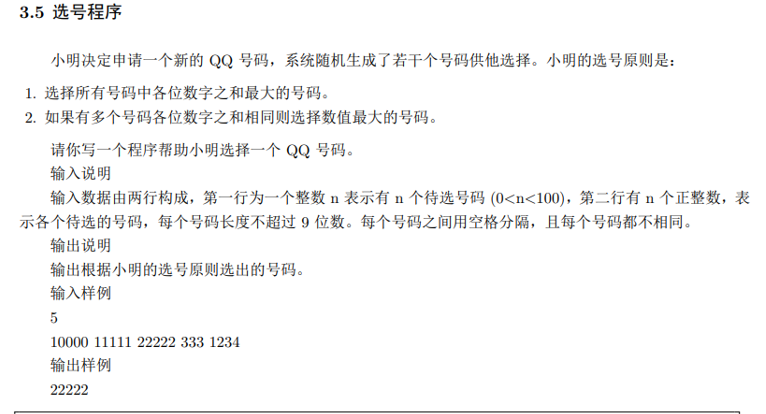
冒泡原理（数组排序）：大的数下沉，小的数上浮

例如，有N个数字，要使他们按照从小到大的顺序输出，则进行N-1趟比较，每一趟比较中选择相邻两项之间的数进行比较，大的数向后排，小的数向前，在第j趟比较中，由于在第j趟时有j个数已经排好顺序了，还剩下N-j个数字需要进行进一步排序，所以在第j趟比较中需要对N-j个数字每相邻两项进行排序，所需次数为N-j-1次，则可利用for循环，定义另一循环变量i，使i<=N-i-1时对每相邻两项比较，将剩余N-j项中最大的排至末尾

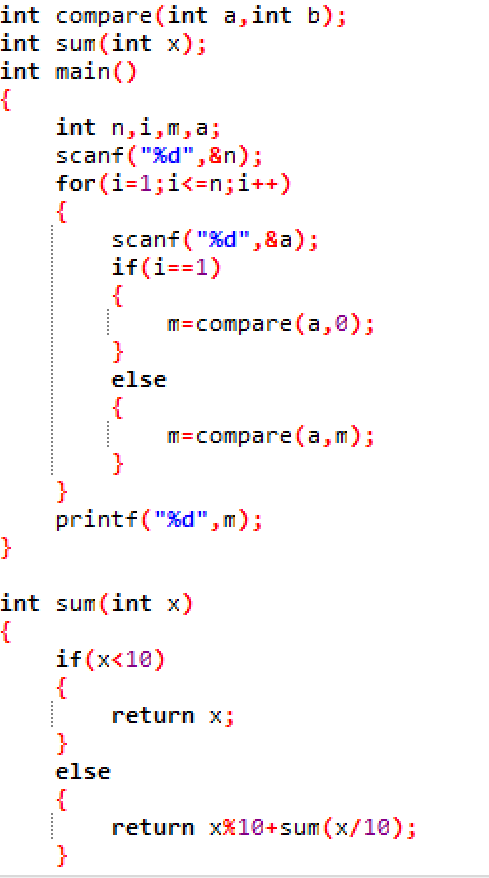
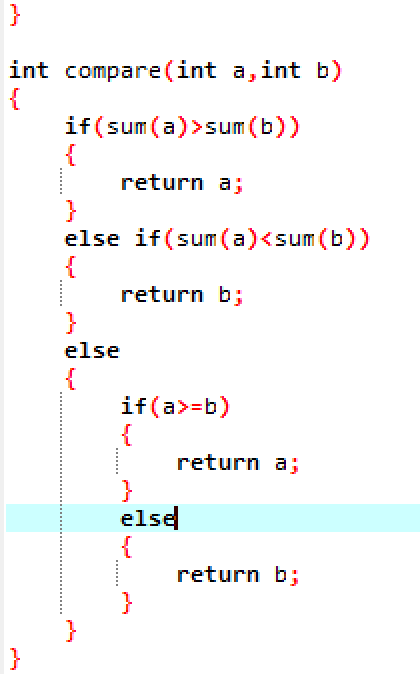
具体操作如下：



上机例题：选号程序



方法一（应用函数）：分别用两个函数使其满足两个对应条件



方法二：

