本文档记录CPrimerPlus一书中所提到的C语言编程注意事项

1. main函数的标准形式（P18）

int main（）

{

return 0；

}

这种形式便于移植

1. 变量的声明位置（P19）

变量的声明位置应 该尽量位于一个代码块的开始。以便于移植到老的机器上也是可行的

1. 变量名字的选择（P20）

变量名字第一个字符必须是下划线或者字母，变量名字可以由下划线和字母以及数字共同组成。

由于操作系统和C库通常采用以一个或两个下划线开始的名字。故自定义变量名尽量不使用下划线作为开始

1. 初始化变量（P37）

初始化变量时

int cows = 29，goats = 14； 这种是可以允许的

int cows， goats = 14； 这种做法不好，容易使人误以为cows也被初始化。

1. 多种整数类型选择（P39）

诸多整数类型中优先选择unsigned类型，没有负数且表示数值范围大

1. 常量（P41、P43、P48）

整型常量：初始化时未指定存储类型时，系统自动为其寻找能满足其大小的存储空间，浪费时间。初始化常量时可自行指定。

如 1000000L系统将自动用long int型的空间去存储该常量。（应该用L而不是l，l容易与数字1搞混）

再比如 2345L 表示用long int的空间去存储2345这个数，而不是按照int去存储。

浮点常量：默认情况，系统将浮点常量当做double存储。

如 some 是个float类型，some = 2.0 \* 4.0；

由于2.0和4.0均采用double类型存储，均为64位，乘积运算采用双精度，结果被截成正常的float 长度。虽然保证了计算的精度，但是浪费了计算时间。

也可采用整型常量的做法，3.45F即使用float类型去存储3.45这个常量。3.45L则表示按照long double类型存储

字符常量：C语言将字符常量视为int类型而非char类型。

如 char grade = ‘B’；

‘B‘作为数值66存储在一个int里面，而赋值后的grade则把66存储在一个8位单元中。

利用此特性可以定义一个字符常量‘FATE’，这是将四个独立的8位ASCII存储在一个32位单元中。如果把这个字符常量赋给一个char变量，那么只有最后8位会起作用，因此变量的值为‘E’。

1. printf的注意事项（P42）

printf（“双引号中有几个格式说明符，后面的参数就得有几个”）参数数量匹配

printf（“引号内部格式说明符表示的类型得和后面的参数一致”）格式一致

如uint16\_t类型的变量，打印的时候格式说明符得是能打印16位的，不能全部使用%d

1. 隐式格式转换（P42）

在short和int类型长度不一致的系统，C在处理short时会自动将short类型转换成int类型。int类型时计算机处理起来最方便最有效的整数类型。

9、可移植类型（P46）

头文件inttype.h中包含有可移植类型的变量别名

如int16\_t 、int32\_t、等

同样包含printf中格式说明符的可移植类型

10、浮点常量（P48）

没有小数点的数不一定不是浮点数

如2E5，使用这种形式就表示其为浮点数

浮点常量可以省略小数点可以省略指数部分，这两者不能同时省略

可以省去小数部分也可以省略整数部分，这两者也不能同时省略。

11、printf的使用（P53）

printf（“adada”“asdwad”“asdwa”）；可以将多个双引号中内容连接在一起打印出来

printf（“只要不在引号内部或者一个单词中间进行断行，使用就没有问题”）

第四章 字符串和格式化输入/输出

12、scanf的使用（P62）

Scanf开始读取之后。会在遇到的第一个空白字符空格（blank）、制表符（Tab）或者换行键（newline）停止读取。所以scanf无法读取带有空格的一连串单词。

13、strlen函数（P63）

strlen函数是以字符为单位给出字符串的长度。不包括结束符\0。但包括字符中的逗号和空格

14、sizeof函数（P64）

sizeof函数使用时对象可以有具体的量或者某个类型。针对某个类型如float、int等时必须使用括号。而针对具体的值时可以不使用括号。但规范的做法是都加上括号。

15、宏定义（P65）

定义明显常量时 不能有等号和分号。如 #define AGE = 20；这样大错特错

应当去除等号和分号。这种宏定义一定要全部大写，增加程序可读性。

16、系统定义的明显常量（P66）

系统在limit.h以及float.h的头文件中分别提供了有关整数类型的大小限制的详细信息。如整型数最大值或者最小值。以及char型或者unsigned char 类型的最大值和最小值。

17、printf的修饰符（P70）

其中有几个需要注意的点

.digit(s) 表示精度。对于%e、%E、%f转换，是将要在小数点的右边打印的位数。

对于%s转换，则是将要打印的字符的最大数量。

对于整数转换，则是将要打印的数字的最小位数。

digit（s）表示字段宽度最小值。如果该字段不能容纳要打印的数或者字符串，系统就会使用更宽的字段。

hh和整数转换说明符一起使用，表示一个char或者unsigned char 类型数值。

h和整数转换说明符一起使用，表示一个short int 或者unsigned short int

18、printf（）的标志（P71）

+表示输出的值带上正负号

-使输出值是左对齐的

#使用转换说明的可选格式。若为%o格式则以0开始；若为%x和%X格式，则以0x或者0X开始。

19、printf（）的返回值（P76）

Printf具有一个返回值，返回所打印的字符的数目。如果有输出错误，那么printf会返回一个负数。

20、打印较长的字符串（P77）

打印较长的字符串有三种方法

1、是使用多个printf语句。第一个打印的字符串不要以\n结尾

2、是使用反斜线/和回车键结束第一行

3、是采用字符串连接的方法。如果在一个用双引号引起来的字符串后面跟有另一个用双引号引起来的字符串，而且二者之间仅用空白字符分割，那么C会把该组合当作一个字符串来处理。

21、使用scanf（P78）

Scanf函数使用空格（换行、制表符和空格）来决定怎样把输入分成几个字段。他依次把转换说明与字段相匹配，并跳过他们之间的空格。如果存在多个转换说明符，实际输入的时候输入与输入之间可以用空格或者制表符什么的间隔。Scanf可以识别出来。

22、从scanf的角度看输入（P79）

假定你采用一个%d说明符来读取一个整数。Scanf（）函数开始每次读取一个输入字符，他跳过空白字符（空格、制表符、和换行符）直到遇到一个非空白字符。因为他试图读取一个整数，所以scanf期望发现一个数字字符或者一个符号。如果他发现了一个数字或一个符号，那么他就保存并读取下一个字符：如果接下来的字符是一个数字，他保存这个数字并读取下一个字符。就这样持续直到遇到一个非数字的字符。Scanf把这个非数字字符放回到输入中。这就意味着下一次开始读取输入时，他将从前面被放弃的那个非数字字符开始。

如果使用了字符宽度，那么scanf在字段结尾或者在第一个空白字处（二者中最先到达的一个）停止。

不能通过字段宽度使得scanf用一个%s说明符读取多一个词的输入。

Scanf不是C最常用的输入函数，一般常用getchar（）或者gets（）

23、格式化字符串中的常规字串（P80）

如scanf（“%d，%d”，&a， &b）

格式化字符串中含有一个逗号，所以你输入的时候也应该有个逗号，否则会出现错误。

输入第一个数的时候后面必须紧跟一个逗号。

此例中逗号后面还有个空格，格式化字符串中的空格表示跳过下一个输入项之间的任何空格。

你这样想。对于scanf中的格式化字符串中的内容每一个东西在你的输入中都得有所体现。比如(“%d, %d”)这样，你第一个就得得输入一个整数，整数后面必须得跟个逗号。逗号后面是个空格，所以你输入的时候什么空格、换行、制表符等等都可以算作格式化字符串中的空格，所以此时用户输入的时候在逗号后面完全可以跟空格、换行、制表符等。

再比如(“%c”)格式化字符中只有一个%c，那么上来无论是空格还是什么东西，他都直接读取输入中遇到的第一个字符。如果使用的是(“ %c”)这种形式，你输入的话第一个得输入一个空格，然后才会读取输入中的字符。所以如果你这样写的话，scanf就会自动跳过前面空白字符，读取第一个非空白字符。

你反正记着，scanf中的格式化说明符；里面的任何东西你在输入里面都得有对应。

24、printf和scanf中\*的使用（P84）

在printf中\*使用表示字符宽度。这个字符宽度可以在程序中由你指定。如果使用\*表示你还需要多指定参数，就是字符的宽度。

在scanf中则是放在%和说明字符之间，他使函数跳过相应输入项

第五章 运算符、表达式和语句

25、除法运算符（P92）

整型数的除法运算符和浮点数的除法运算符有很大差别。浮点类型的除法运算得到一个浮点数结果。而整型数运算得到一个整数结果。整数不能有小数。整数除法的小数部分都被丢弃，这称之为截尾。

正常情况下应避免混合类型的运算。对于负数的截尾，C99之后-3.8截尾成-3.

26、运算符的优先级（P93）

如果两个运算符优先级相同，且共享一个操作数，会根据他们在与剧中出现的顺序不同执行他们。对于大多数的运算符，改顺序是从左到右的。（=是一个例外）。

如果两个运算符优先级相同，但是不共享一个操作数，如y = 6\*12 +5\*20；

这个时候，计算机就将运算顺序由程序员决定。可以通过括号来决定谁先谁后。哪一种方式最适合当前的硬件就用哪一种。

27、sizeof运算符（P95）

sizeof运算符是以字节为单位返回其操作数的大小。（字节的单位是根据char类型的大小定的）。C规定sizeof返回size\_t类型的值。这个size\_t类型是C语言通过typedef定义的。如同int32\_t这样的，都在头文件里面定义。同样的，在printf中，也有专门针对size\_t的格式说明符%zd。

28、取模运算符%（P96）

不能对浮点数使用取模运算符。取模运算符只能对整型数使用

取模运算符的第一个操作数为负数，则最后的结果为负数。如果为正数，则最终的结果为正数。

29、增量运算符（P98）

增量运算符有两大优点，一是使程序更加整洁，增加可读性。二是增量运算符通常产生更高效的机器语言代码。因为它与实际的机器语言指令类似。

增量运算符使用过程中的注意事项，应当避免在程序中使用那种前缀形势和后缀行驶不同效果的代码。

例如，不要将赋值=和增量预算符++一起使用。形如b = ++i；这种形式。

增量运算符和减量运算符具有很高的运算级别，除了圆括号比他们高之外，就没有比他们运算优先级更高的。

增量运算符的使用原则

1. 如果一个变量出现在同一个函数的多个参数中时，不要增量或者减量运算符用在它上面。
2. 当一个变量多次出现在一个表达式里时，不要将增量或减量运算用在它上面。

30、表达式和语句（P102 P105）

表达式是由操作数和运算符组成的。最简单的表达式只有一个变量或者一个常量而没有运算符。

C的一个重要属性就是每个表达式都有一个值。

语句是一条完整的计算机指令，在C中，语句用结束处的分号标识。语句是构成程序的基本成分。有简单语句和复合语句之分。语句包括：

声明语句、赋值语句、函数调用语句、结构化语句（while（toes>2） toes = toes +2；）

空语句；复合语句或者称之为代码块（block）。

31副作用和顺序点（P103）

从C的角度看，他的主要目的是对表达式求值。每个表达式都有个值。对于一个赋值语句，C的主要目的是计算表达的值，其副作用是给这个变量赋值。

一个顺序点是程序执行中的一点；在该点处，所有的副作用都在进入下一步前被计算。在C中，语句里面的分号标志了一个顺序点，他意味着在一个语句中赋值运算符、增量运算符、以及减量运算符所做的全部改变必须在程序进入下一个语句之前发生。并且任何一个完整的表达式的结束也是一个顺序点。

完整的表达式是这样一个表达式，他不是一个更大的表达式的子表达式。

如while（guests++>10）

Printf(“%d”,guests);

While的循环条件判断是一个完整的表达式，所以在进入下一个完整表达式之前，增量运算符的计算应该已经完成，故printf所打印的变量值应该为增加过后的变量值。

32、类型转换（P105）

类型转换有以下几条原则

1. 当出现在表达式中时，有符号和无符号的char和short类型都能自动转换为int，在需要的情况下，将自动转换成unsigned char （如果short 和int具有相同的大小时）反正按照能满足条件下的最高标准转化。那个级别更高就转化成那个。
2. 当做为函数的参数被传递时，char 和short会被转换成int 而flooat则被转换成double

数据类型的提升往往是平滑且无害的，但是降级会导致真正的问题。写程序时宁可数据类型提升也不要让数据类型降级。