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