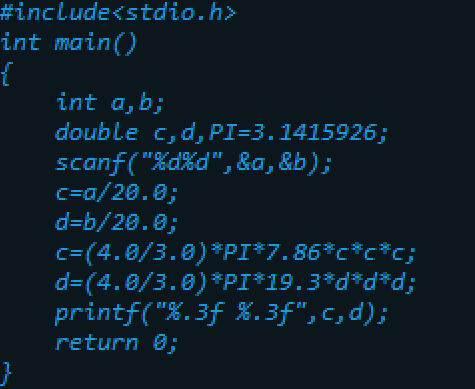
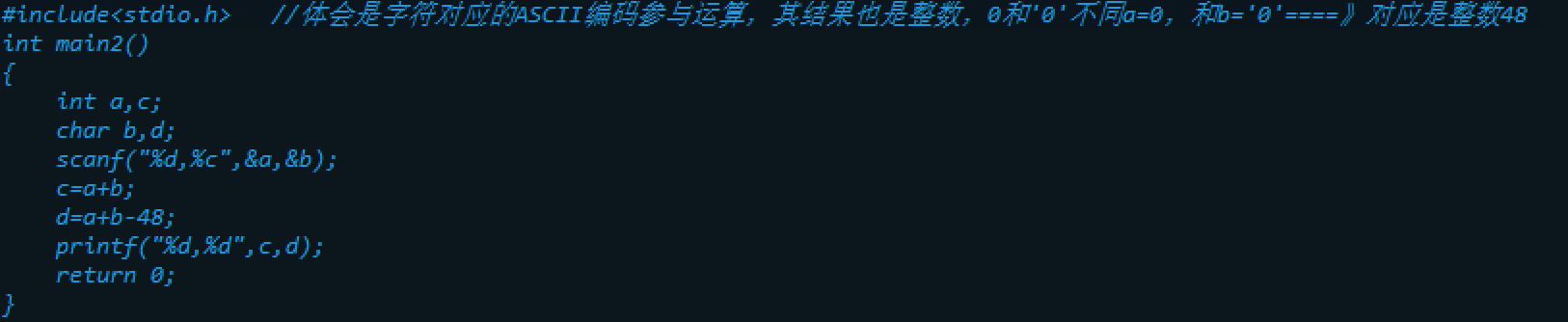
计算机导论与程序设计学习报告

进入了大学之后，是我第一次这么系统的地接受计算机学习，感觉很新奇也很感兴趣，新奇于C语言的严谨和运算魅力，感兴趣于运行的过程和结果。

入门c语言，我懂得了#include<stdio.h>是编译预处理指令一切的程序都要由它开始，也知道了int main()是定义主函数的意思，在多次的实践中也清晰的记得了int main()后的括号千万不能忘记，我也明白了一个函数的开始标志是“{”，结束标志也对应了“}”知道了printf是输出的意思，还有最后的return 0;不能少，还有最让我影响深刻的是“;”，一次又一次的运行失败和红色区域的出现让我不得不记住分号的的重要性对于语句和整个的程序的重要性，我还学会了集中数据类型，有整数型，还有包括单精度浮点数和多精度浮点数的浮点型数据，以及字符变量等等，他们截然不同，有时如果一不小心用错变量会让整个程序的结果完全不同。

* 以第一次上机的一道题目为例，【计算球的重量】，关于c=a/20.0而不是c=a/20的细节问题，若为20的话，a是整型函数，20也是整型函数，整型函数除整型函数的结果也是整型函数，结果若为整型函数就会影响后面的结果，所以后面一些相应的计算都加上了0.0的精度，这个程序也让我应用了取精度的用法%.3f。



1. 第一次上机也让我更加清晰了ASCII编码的概念【数字字符】

字符的‘0’和整型函数的0是不同的两个不同的概念，字符‘0’对应的ASCII编码是整数48，参与运算的结果也不一样。

1. 第二次上机复习了switch-case的用法
2. 当case是符号时要用case ’+ ’,若是数字则是case 1。
3. break是跳出整个循环。
4. 还有关于取余和取整的应用。
5. Switch后紧跟大括号

{

case 1：

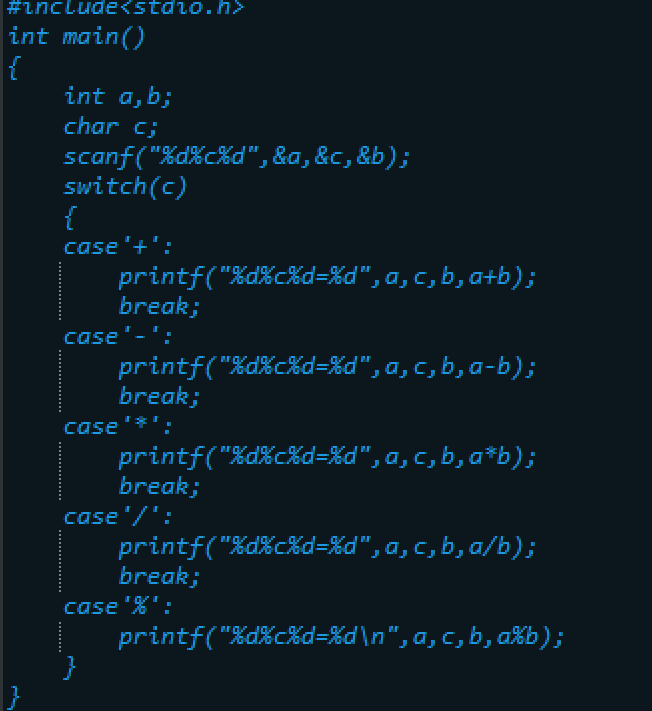
break；

case 2：

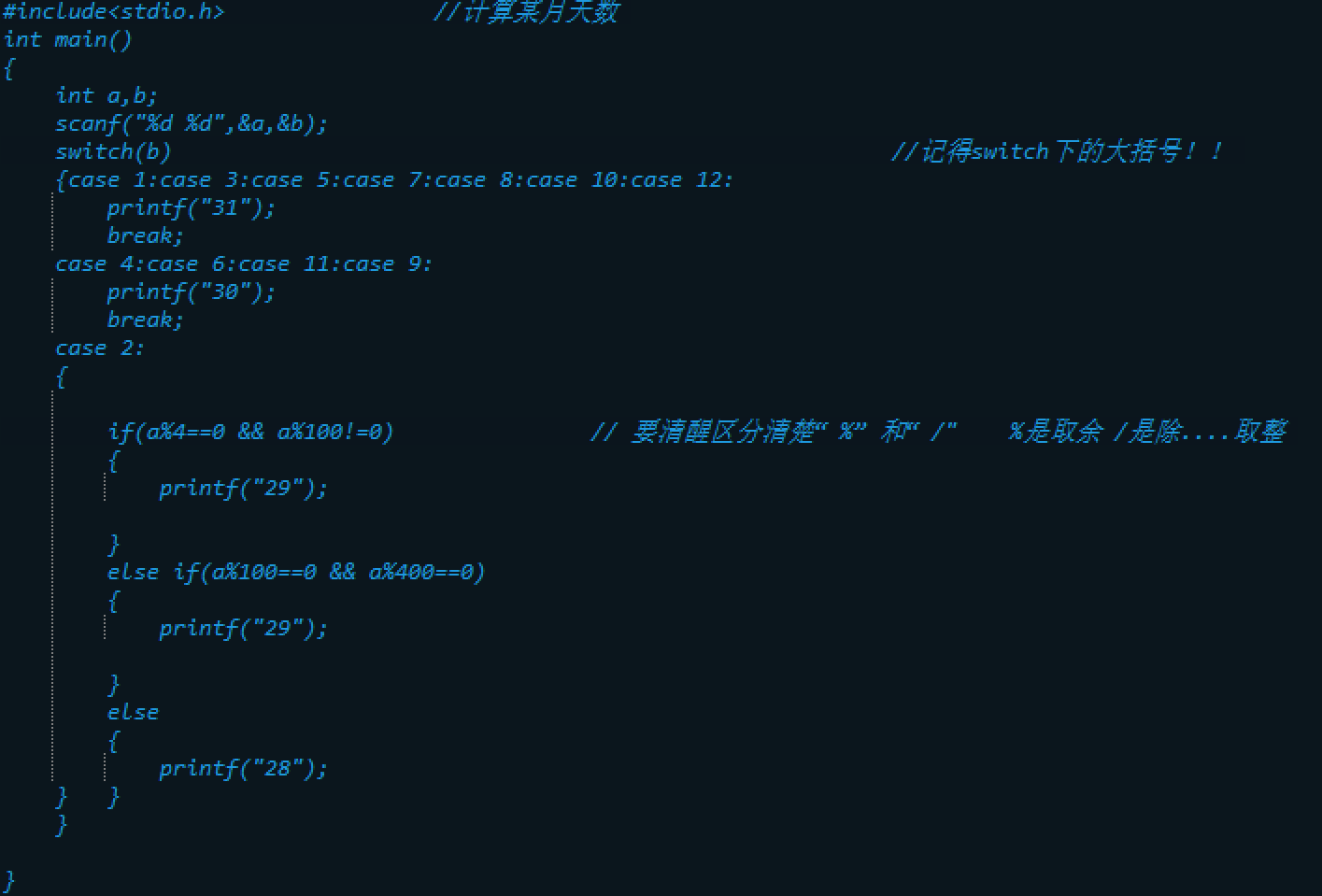
break；

}

【四则运算】



【计算某月天数】

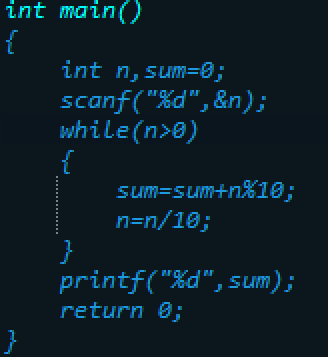


1. 关于数学的范围和C语言范围的区别

数学的表达 2<=a<=3;

而c语言的表达 a>=2&&a<=3;(若写成2<=a<=3的话是先判断2<=a为真的话结果为1，1<=3也为真)

1. 【计算整数各位数之和】：关于取余和取整的灵活应用

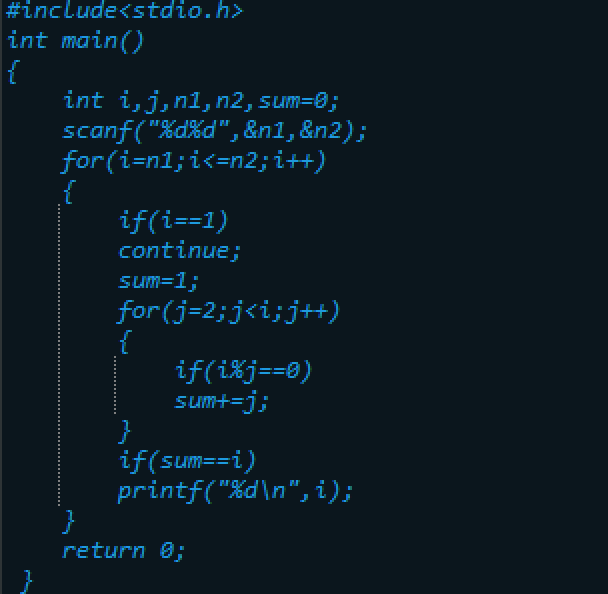


逻辑循序：while括号里是运行while后大括号的条件，用条件来控制，当n<=0时循环结束，n%10是得到个位数，n/10是去掉个位数，然后依次得到每一位数相加，得到各个位的和。（当时做这道题有一个卡住的点就是没想到要运用while循环，首先这道题它数字的位数不知道，这就无法让我们一位一位的算出来再相加，而应该用while循环来解决这一问题。

1. 关于内层循环sum的初始化问题

【完数】

1总是一个整数的合法因子，所以sum要初始化为1！



1. 最大公约数的一题多解
2. 辗转相除法（用到了if-else和while）

注意点

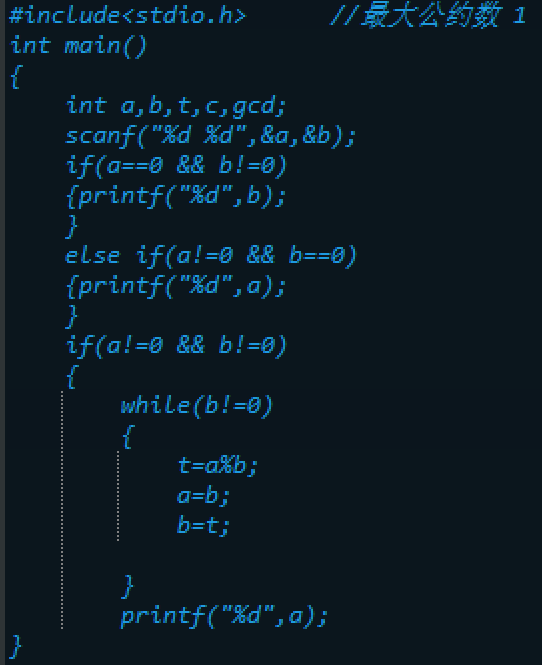
1. 要考虑a和b是否为零的情况
2. 辗转相除法

t=a%b；

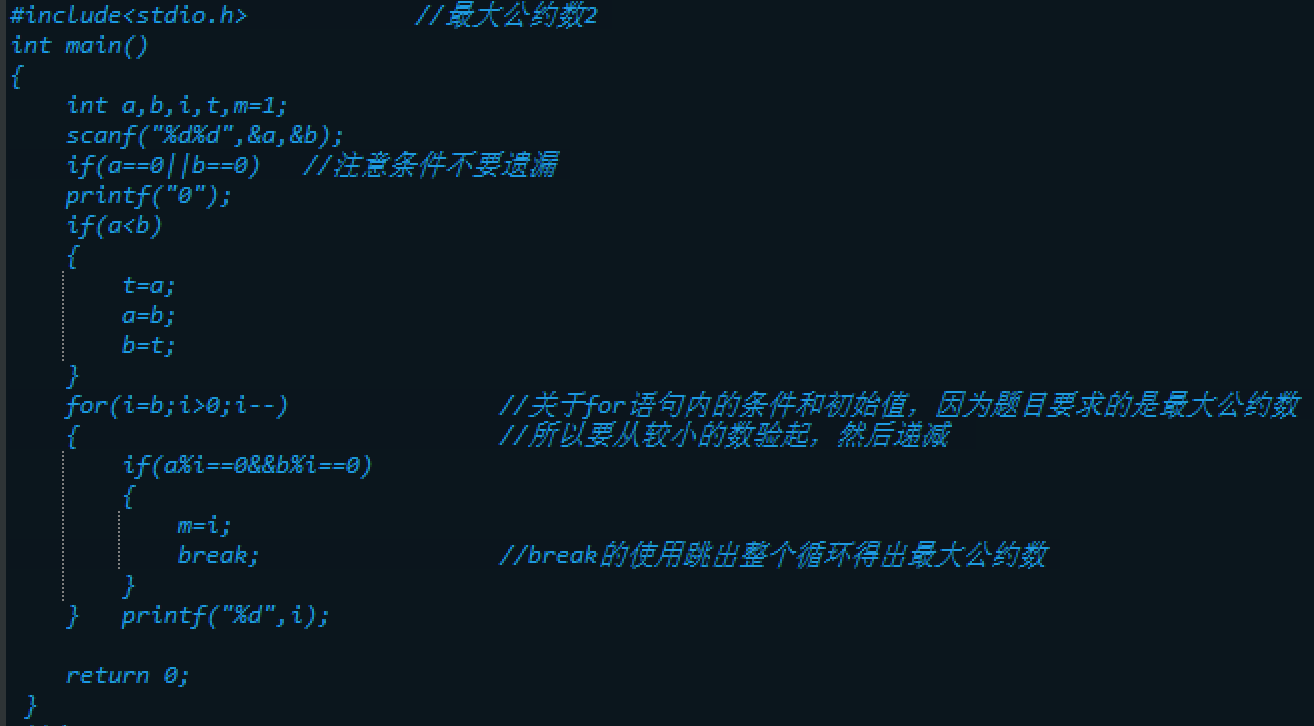
a=b；

b=t；

1. 注意循环终止的条件b=0时表示整除，此时就应该跳出循环

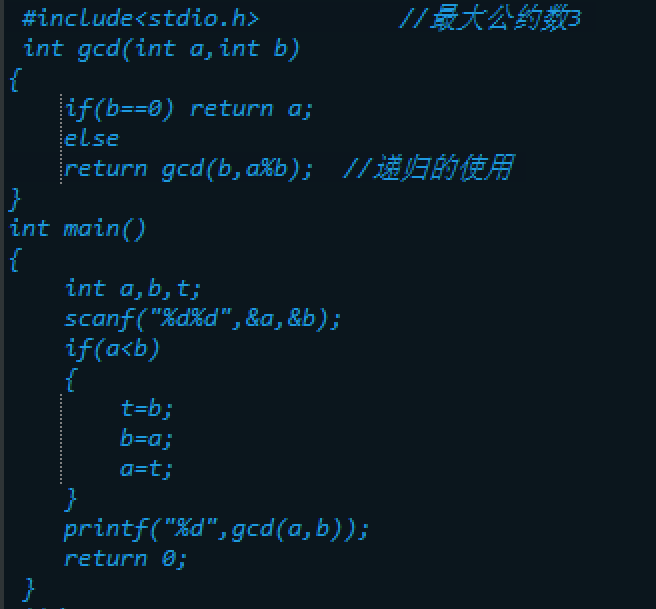


1. 通过一次一次循环找到最大公约数



注意点

1. 先比较大小，因为要用较小的数开始递减以寻找最小公约数
2. 关键点：i的初始量应为b，然后i—
3. 条件a和b都能整除i时就是最大公约数，运用break跳出整个循环（continue是跳出本轮循环）
4. 调用函数求最大公约数



1. Int gcd（int a,int b）是自定义的函数，可以运用到后面的运算作为函数的调用
2. 递归的运用return gcd（b，a%b）?
3. 【车牌限号】

(a==1&&(d==1||d==6))

关于条件判断的先后和并列条件，加上括号表明顺序。

* 关于循环:key: for, while, do while

1. While循环有一点非常重要，在构建while循环时必须让测试的值有变化，表达式最终要为假，也可以用break和if语句来终止循环，while循环的入口为满足条件的情况，while循环的出口为不满足条件时。
2. For循环

For（初始表达式；条件表达式；改变循环变量）

{

（循环体）

}

满足条件表达式时执行循环体，不满足时跳出循环

1. While循环和for循环都是入口条件循环，do while是出口条件循环。
2. do while循环与for和while的区别是do while是再执行完循环体后才根据测试条件决定是否再次进行循环，也就是说do while不需要满足任何条件就可以开始第一次循环至while在判断下一次循环，所以至少进行一次循环，而for和while可能不进行循环
3. 循环语句的选择:

当涉及初始化和更新变量或者计数循环for(count=1;count<100;count++)时，用for循环比较合适，其余情况用while比较合适。

1. 三种循环都可以用continue和break语句，注意break可以用于循环和switch语句中而continue只能用于循环中

* 分支和跳转：key：if ，else ，switch-case ，break ，continue

1. 注意在if和else之间执行多条语句时，必须用花括号把这些语句括成一块
2. 多重选择

* If（）

{}

Else if（）

{}

else if（）

{}

……

Else

{}

* Switch和break

1. Else和if的配对问题

在无花括号时else和距离最近的if匹配

在有花括号时else与花括号外距离最近的if相匹配

1. Switch和if else的选择

Switch用于有多种情况每种情况都有明确的值

If else更适用于浮点类型变量和表达式 such as：if（a<100 && a>0）

* 关于一些易错点整理
* 定义数要确定清楚它的类型
* Scanf不要漏了“&”
* For，if括号后没有;
* Do while语句的while();后有;
* 关于精度问题可以人为使用1.0来调整
* 区分“==”和“=”，“==”是判断是否相等，而“=”表示赋值
* 引入函数运算要记得加上#include<math.h>

我知道了每一次成功的运行背后或许都有着许多次的错误，但我也对每次的错误都心怀敬畏，正是因为这一次次的错误，我才能一次次的进步，才能在每一次编写时都能减少一次又一次的错误，我想这就是错误的珍贵价值吧，我们作为一名合格的大学生，不能只对于会用一些简单的软件而沾沾自喜，而应该学得更加的深入，对自己有更高的要求，不能只流于表面，要学会系统化和专业化地学习计算机的语言，计算机语言是人与计算机交流的工具，能让我们进行程序设计，编写出合适的程序，能让我们解决许多单靠人力无法解决的，对于计算机语言的学习能让我掌握一项新技能，也能让我更加懂得计算机的运行。