我主要存在以下的这些缺点：

1、 学习耐心与细心不足，如scanf(“%d”,&n)；中的“&”有时候会忘了。而在最后输出时又错写成printf(“%d”,&n);从而错误得输出了地址而不是我原来想要的答案。

2、 编程思想不够发散，看着题目有时想不出解答的方法，更不用说编写程序来解题了。

3、 基本功不够，有些函数的表达不太精通，需要看书来核实，以致耗时较多。 知识不够广，有些内容没有学好，不能要用到时及时反映出来，认识程度不够深刻。

4、 有时候不够精简，有一点用处不大或者说没有也可以的文字存在。 英语水平较差，对错误的地方虽然电脑有说，但由于是英文，理解上还是存在一点问题。

5、 为了能更好地学好C语言，在今后学习中我要更多的动脑，综合运用所学，多看相关东西，多上机练习，提高电脑水平，增强自学能力，把已会的东西掌握好。

第一周上机

第一个程序：计算球体重量#include<stdio.h>int main(){ double fe=7.86,au=19.3,PI=3.1415926,Mfe,Mau; int Rfe,Rau; printf("Rfe是："); scanf("%d",&Rfe); printf("Rau是："); scanf("%d",&Rau); Mfe=PI\*Rfe\*Rfe\*Rfe/1000\*fe/6; Mau=PI\*Rau\*Rau\*Rau/1000\*au/6; printf("%.3lf %.3lf",Mfe,Mau); return 0;} 首先结果需要是小数点后三位实数，所以结果应定义为实数且””内结果应%后加.3。其次是知道算式可以简化是尽量简化，有一个同学Mfe和Mau这里用了巨多括号，结果还错了。尽量用double,因为float差不多只精确到小数点后6位；如#include<stdio.h>int main(){ float a=3.1415926; printf("%f",a); return 0;} 结束是3.141593； 在定义pi时可以在开始定义也可以在(float pi=)定义。

在scanf(“”)的使用时注意不能使用（\n）进行换行

第二个程序：温度转换#include<stdio.h>int main(){ float T1; double T2; printf("华氏温度是："); scanf("%f",&T1); T2=(T1-32)\*5/9; printf("摄氏温度是：%.2lf",T2); return 0;}关键就是这个整数问题，整数除以整数还是整数；当我把T1定义为int时，由于(T1-32)\*5/9都是整数，T2也必然是整数；

注意除法的运算法则注意添加小数点避免计算法则错误而导致结果错误。

第三个程序：数字字符#include<stdio.h>int main(){ int a(0<=a<=4); char b('0'<=b<='5'); printf("请输入整数a与字符b\n"); scanf("%d,%c",&a,&b); printf("%d,%c",a+b,a+b); return 0;} 这个最有趣的就是这个scanf里的逗号，不加就错，因为逗号也会算进去(因为b是字符，看第五个程序)； 还有就是写这个范围好像没什么用，写个a=5还是会算；

注意字符的定义与输入输出的代换。

上周本次上机题目车辆限行问题

#include<stdio.h>

int main()

{

int a,b,i,n,c,m;

scanf("%d %d %d",&a,&b,&i);

n=a%10;

c=n;

m=n;

if(i<200)

{

printf("%d yes",n);

}

if(200<=i&&i<400)

{

switch(b)

{

case 1:if(n=1||n=6)

{

printf("%d yes",c);

break;

}

else

{

printf("%d no",c);

break;

}

case 2:if(n=2||7)

{

printf("%d yes",c);

break;

}

else

{

printf("%d no",c);

break;

}

case 3:if(n=3||8)

{

printf("%d yes",c);

break;

}

if(n!=3||8)

{

printf("%d no",c);

break;

}

case 4:if(n=4||9)

{

printf("%d yes",c);

break;

}

if(n!=4||9)

{

printf("%d no",c);

break;

}

case 5:if(n=5||0)

{

printf("%d yes",c);

break;

}

if(n!=5||0)

{

printf("%d no",c);

break;

}

case 6:case 7:

printf("%d no",c);

break;

}

}

if(i>=400)

{

switch(b)

{

case 1:case 3:case 5:

if(m==1||m==3||m==5||m==7||m==9)

{

printf("%d yes",c);

break;

}

else

{

printf("%d no",c);

break;

}

case 2:case 4:

if(m==0||m==2||m==4||m==6||m==8)

{

printf("%d yes",c);

break;

}

else

{

printf("%d no",c);

break;

}

case 6:case 7:

printf("%d no",c);

break;

}

}

return 0;

}

由源代码就可以知道由于思维不是很清晰很少使用循环语句导致了代码过长，书写时间过长表达内容重复等情况的出些，在后续的学习中应该注意逻辑思维的训练和一些易错点的加强

其中if语句的条件判断句刚开始时由于或语句习惯思维导致错误无法得到理想的结果，最后通过耐心调试了==和m的判断才解决问题。

Int main()

{

int i, a[10];

for(i=9;i>=0;i--)

a[i]=10-i;

printf("%d%d%d",a[2],a[5],a[8]);

Return 0;

}

其中定义a[10] 时，其中的10表示数组长度不能用变量表示例如a[n]这样的表达时不正确的，而a[i]=b中的i表示的是下标可以用变量表示a[i]=b这样的赋值语句表达正确。

另外注意下标是重0开始这就为我们的循环埋下了出错的可能。由下标命名规则可知判断语句中i最大取值为i-1.

实验中我深刻意识到完成程序的编写，决不意味着万事大吉。认为万无一失的程序，实际上机运行时可能会出现很多意想不到的问题。有时编译程序检测出一大堆错误，有时程序能够顺利运行，但是运行结果并不是你预期中想要的。因为开发环境所提供的编译系统无法发现程序逻辑错误，或者是你原来所设计时的理论错误，这就只能靠自己的上机经验来分析判断错误的所在了。所以程序的调试是一个技巧性很强的工作，它可能比编一个程序耗时更多。由此可看出上机实践的重要性。