**1（大一时候写的总结，可能有不当的地方，课上会一并解释）**

**第一部分 数据类型**

1.无符号数被附负数时,会将负数的补码存入内存,但在之后计算中,符号位被当做数值计算

例如unsigned int i;

i=-9;

printf("%u",i); 输出后是4294967287

本来首位的1只是符号,但现在却用来表示数的首位

（可以拓展到内存模型）

1. 字符串中的特殊字符必须用转义字符表示,如’\’必须写做’\\’,否则将被视为转义字符,无法输出.
2. printf函数与scanf函数的区别:
3. Printf函数的格式字符一般应与表达式值的数据类型相同(但不强制，不强制但会出错，比如2.3用%d输出，指数部分会被认为是数值，成161062799)，但scanf强制,否则无法输入。

同理,地址变量类型也应与格式字符一致.

例如:不能scanf(“%d”,&(float)a); 然后输入2.5,这样无法输入(直接报错弹出)

1. Printf中输出无论是单精度浮点数还是双精度浮点数都可以用%f输出，但scanf单精度用%f，双精度用%lf。
2. Scanf需要取地址符&
3. 总的来说scanf要求更高。
4. scanf函数时键盘输入数据的类型要与格式字符相容，否则不仅当前输入的数据无效，后续的输入也不再被处理(主要是不能识别,就卡在那了,比如3.5与%d,第1个scanf接收了3，第2个遇到 .5拒绝接收,就一直卡着挡着后面数据接收)
5. 注意在getchar()函数中提示过的问题，scanf("%c")也会接收之前输入但未被处理的任何字符
6. 只有完整输入scanf中的内容，才能在每个地址存入正确的数据，少一个字符都会因为输入格式不符而直接跳过整个scanf的输入
7. int整数除法致命,如果不进行类型转换,就会截断取整(切记切记)
8. 键盘输入数据再存入内存的数据转换方式比较玄学(应该与编译器有关)

例如:scanf(“%d%d”,&a,&b) 若输入2.3 会显示a的值为2,且b的值无法输入??

并不是直接把补码塞进.

1. !!z把任意数变成1(可能有用)
2. 赋值运算符=使用的数据类型转换比较强硬,不管谁的表达能力强,就是右边向左边看齐
3. 在隐式类型转换中,有”整形提升”规则:

对于所有比int小的类型(short,char,unsigned char等)首先会提升为int型,再去计算(这里的计算不包括赋值)

1. **不要把赋值号=与等于号==弄混用错!!!!**
2. If只是个顺序语句,会导致以下结果

例如:

if(f\_a == 0)

{

printf("输入的二次项系数为 0，方程不是一元二次方程。\n");

return(10);

}

此时return语句会将10返回到main函数,而不是if(不是函数,无值),而这会导致从此往后的所有语句不再计算,main函数结束(已经返回).

还包括break:

For()

if(a == 0)

break

会直接跳出循环

14.%\*d指跳过某个输入的内容 使下一个输入的内容复制给紧接着的变量。

例如下面程序 输入 1 2 3 则输出为 1 3 程序把2跳过了，直接把3复制给b。

int a,b;

scanf("%d%\*d%d",&a,&b); //这里中间的%\*d让程序忽略第二个输入。

printf("%d%d",a,b);

15.回车的ascii码是10,空格是32.

16.逻辑等价性:（逻辑代数）

例如:判断闰年

!(year%100)&&!(year%400)||year%100&&!(year%4)）

有很多等价写法，，，

1. char与int除了字节数外几乎完全一样,不同在scanf上

若用%d时去输入a,什么都输入不了,因为输入数据与格式字符不一致;

Int a

Scanf(“%c”,a)

（四个位置都能存）

1. 希望输出的数字不连在一起?

法一: for (i=0;i<30;i++) {

if (i%5==0) printf("\n");

printf("%12ld",s[i]); }

换行与字符长度限制结合使用

法二:for (i=1; i<=20; i++)

printf("%d%c", ncow[i], i%5? ' ':'\n');

使用条件判断,选择空格或换行

1. 二维数组对应二层嵌套循环(都是先固定行数,再扫描列数的)
2. 扫描和遍历的概念 (for循环,一个一个来判断)
3. 声明变量时一定要思考变量的数据类型(以后计算时要想着类型,否则可能会隐式类型转换造成截断取整)和是否赋初值(不赋初值可能不是0而是系统随机数)
4. For循环如果是与数组配合,循环控制变量i的初值尽量为1,且数组的大小多设一位,这是因为有时候处理有关数字的问题时,i最好直接显示这个数字,而不要差个1,导致混乱
5. Getchar一次只能接一个字符,所以输入abcd,也只会储存a
6. 类似

CHAR\_MIN,CHAR\_MAX的形式可以求各种数据类型与函数(像rand)的取值范围

Ps:要加上头文件stdlib.h

1. %.0f会把输出的浮点数先四舍五入(不是取整)至个位再输出.
2. Scanf时输入0121,系统只会接收121,而不是八进制数(只有编码时才会认为是八进制数
3. scanf("%s")不会吃空格，后面需要一个getchar()
4. int a[n]不是个好习惯，因为本来就是错的

**第二部分 函数**

1. 有两个地方容易忘记加上分号:

一是do-while后的while后要加分号

while(a>b);

二是函数原型声明要加分号

Long factorial(int a);

2. 函数原型声明放在程序头部是自然的,因为头文件里也是函数原型声明,放一起很和谐.

C++里很强调头文件的作用

3. 全局变量与局部变量可以同名，但本质上是两个不同的变量，且在局部区域内全局变量无效，只有局部变量有效，称作定义屏蔽

4. 在main函数中任何地方使用return语句都会退出整个程序

5. Return除了能返回值外,还能在多出口多分支程序中显示到底是哪个出口

6. Fabs与abs函数都是算绝对值的,只是fabs函数输出浮点数,abs函数输出整形数

7. 函数原型声明中的形参名可以与定义中不同,甚至不写都可以

Int Factorial(Int);

8. 只写一个函数名,比如单独写一个main

printf("面积是%p\n",main); 就会输出地址,其中格式字符为%d是以十进制输出地址,%x以十六进制(非定长)输出,%p是以规定长度(4gb内存为8位数字)输出十六进制

9.循环条件如果是开区间，一定要注意可能溢出有意义数字的范围

例:for(i=0;s[i]<k;i++) 来寻找k在数组s中的哪两个数之间，一定要注意k比全部数组数都大的情况（开区间溢出），如果不加以限制，就会有问题

10.伪随机整数一个通用的公式是：

要取得[a,b)之间的随机整数，使用（rand() % (b-a)）+ a （结果值将含a不含b）。

在a为0的情况下，简写为rand() % b。

最后，关于伪随机浮点数：

用rand() / double(RAND\_MAX)可以取得0～1之间的浮点数（注意，不同于整型时候的公式，是除以，不是求模）

11. 形式参数不能是常量, 且形式参数之间必须用逗号隔开

12．介绍一个条件编译功能 #ifdef a