

一.错误和警告

1. 错误：编译器不会继续进行编译。
2. 警告：来自编译器，尽管编写的代码有效，但可能不是程序员想要的。

二.变量和常量

1. 常量：在整个程序运行过程中没有变化的量叫常量。
2. 变量：在运行期间可能会被改变或赋值的量叫变量。

三.数据类型关键字

1. 字(word)：字是设计计算机时给定的自然存储单位。
2. 字长(word size)：指一次可以处理的二进制数字数目，是CPU的主要技术指标之一。
3. 整数：关键字：int（用来表示最基本的整数类型），short,long,signed,unsigned（这几个用来提供基本整数类型的变式）。
4. 浮点：①浮点数的存储：将浮点数分为小数部分和指数部分来存储（符号，小数和整数）②e指数法：3.16E7表示 3.16×10^7 （不过要注意在计算机中是使用二进制和二的幂进行存储）。③任何区间都有无数多个浮点数，所以计算机的浮点数不能表示区间内的所有值，浮点数通常知识实际值的近似值。
5. 布尔值：用关键字_Bool声明。
6. 复数和虚数：分别用关键字_Complex和_Imaginary表示。

四.C语言的基本数据类型

1.int类型：

1. 占用内存的大小：一般而言储存一个int要占用一个机器字长。如早期16位IBM PC兼容机用16位来储存一个int值，后来PC成了32位，就用32位存储。
2. ISO C的规定：ISO C规定int的取值范围最小为-32768~32767。
3. 声明变量：int XXX;或int XX,XXX;也可以使用多行。**声明的作用：创建并标记存储空间，并为其指定初始值。**
4. 初始化变量：既可以在声明时初始化也可以在后面完成。
5. 打印int类型的值：使用**转换说明%d**，**要确保转换说明的数量和待打印值的数量相同，不然就UB。**
6. 八进制和十六进制数：有些时候我们要使用八进制和十六进制的数更加方便一点，对于字面量，我们使用0x或0X前缀表示十六进制常量，用0前缀表示八进制常量。要将保存的数字以八进制的格式显示，使用转换说明%o，十六进制:%x,如果

要显示前缀，可以使用其他转换说明：如%#o, %#x, %#X（就是在百分号和字母之间插一个#）。

7. 关于其他整数类型：①使用long类型可以获取更大的内存空间，但运算速度减慢（空间换时间）。②**如果在long 和 int 类型占用空间相同的电脑上编写代码，应使用long, 以便与16位机兼容。**③ong long类型的出现时为了符合64位处理器的出现。

8. long常量和long long常量：①一般而言，程序中使用的字面量常量都被编译器当int储存，除非超过了int的表示范围。此时编译器会尝试用：int -> long -> unsigned long -> long long -> unsigned long long类型（不知道为什么跳过了unsigned int）。②如果想要把一个小数字当long类型对待（**比如在编程时要显式使用IBM PC上的内存地址**），可以加一个L或l（不推荐，可能与数字1搞混）后缀，同样的，long long 类型可以使用ll或LL，unsigned long long 可以使用ull, LLU, ULL等。

9. 整数的溢出:UB

10. 打印short, long, long long, unsigned类型：这个比较简单，short---%hd, long---%ld, long long ---%lld（不同于后缀，在转换说明中只能使用小写）,unsigned---使用“u”代替%d中的d。

11. 使用正确的转换说明。

2.char类型

1. **本质**：char类型本质是整数类型，因为从技术层面上看它存储的是整数而不是字符。计算机使用编码来处理字符，即用特定的整数表示特定的字符。（在我看来，编码是字符到整数的映射，解码则反之）。

2. **有符号char和无符号char**：根据编译器的不同char可表示的整数范围有-128~127，也可以是0~255。但不管在哪种环境之下，使用signed char都表示有符号字符型；unsigned char都比傲视无符号字符型。

3. 声明char类型变量：char a;或char a,b;

4. 字符常量：使用**单引号** “ ” 表示一个字符，使用**双引号** ""表示一个字符串。

5. 非打印字符的表示方法：①使用Ascii码：char a = 7;②使用转义字符或者叫做转移序列（escape sequence），比较重要的有“\b”——退格,“\f”——换页,“\n”——换行,“\r”——回车,“\\”——双引号（经常被用于Windows下的文件路径）等。③使用特殊的转义字符：十六进制和八进制，“\7”,“\07”,“\007”都表示为八进制下7所代表的字符。当然还有十六进制，要使用“\x”前缀。

6. 打印（print）char类型：使用转换说明（conversion specification）%c，printf函数中的转换说明决定了数据的显示方式，而不是数据的储存方式。

3.float, double, long double类型

1. 四种浮点数的表示方法：①一般计数法（如0.1，100）②科学计数法（基本上不用于计算机）③e指计数法，e后面代表10的次数（如1e100，.1e-10等）④p指计数法，C99后开始支持，p前代表十六进制数（记得加上0x或0X），p后代表2的次数（如0xa.1fp10代表 $(10 + 1/16 + 15/256) * 1024$ ）
2. 实现浮点数的表示方法：一般都是使用IEEE指定的标准。
3. 三种浮点类型辨析：①**float**: C标准规定至少能表示6位有效数字，通常一个浮点要占用**32位**，8位用于存储指数的值和符号，剩下24位表示尾数。②**double**: 至少能表示10位有效数字，通常一个double要占用**64位**而不是32位，不同的编译器对64位的分配不同，导致结果也不同（有的能表示的范围更大，有的能表示更高的精度）。③**long double**: 这个用于表示比double更高的精度，不过C标准只要求其精度和double类型的相同。
4. 浮点常量的书写格式（以e指计数法为例）：①符号可以省略。②大部分情况下小数点不能丢，特殊情况如2e5这种利用浮点来表示更大范围的整数的情形。
5. 编译器相关：编译器一般把浮点类型常量当作是**double**类型的精度。要改变这种默认设置，可以加后缀f或F（当作float类型），也可以加L（当作long double）或l（不推荐）
6. 浮点值的打印：①打印成一般计数法：**用%f打印float和double，用%L打印long double类型。**
②打印成e指计数法：用%e打印float和double,用%Le打印long double
③打印成十六进制格式：用%a和%A以及%La
7. 关于浮点的一些问题：
①overflow: 超出表示范围，以前是UB，现在显示inf或infinity或其他有无穷含义的内容。
②underflow: 数字太小，导致计算时损失精度。
③NaN: 表示一个无意义的数（这个得根据具体情况来）
④舍入错误: 缺少足够的小数位来完成运算。

4._Bool类型

_Bool类型实际上也是一种整数类型，但原则上它仅占用1位存储空间。

5._Imaginary和_Complex类型

1. 三种复数类型: float _Complex, double _Complex, long double _Complex（以float _Complex为例，他的虚部和实部表示的都是浮点类型量，其他

的同理)

2. 三种叙述类型: float _Imaginary, double _Imaginary, long double _Imaginary
3. 相关头文件——complex.h: ①可用imaginary代表 _Imaginary,用complex 代表 _Complex
②可以使用I (大写的) 代表-1的平方根。

6.特殊的整型：可移植类型

1. 存在的意义：确保相同的类型名在不同的系统中维持相同的功能。
2. stdint.h:
①exact-width integer type:如int32_t, 编译器会将此作为占用32位内存类型的别名。
②minimum width type: 如int_least8_t, 作为可容纳8位有符号整数类型中宽度最小的类型别名
③fastest minimum width type:如int_fast8_t, 其作用如名称所示。
④最大整数类型：有符号intmax_t和无符号uintmax_t, 这可能比long long可表示的范围还要大。
3. inttypes.h: 用于提供以上类型的conversion specification, 如PRIld32就是包含32位整型的字符串宏。

五.类型大小

1. sizeof关键字:利用sizeof关键字可以用来判断指定类型的大小, 使用方法sizeof int或sizeof(char)
2. sizeof的返回类型是size_t类型, 可以使用%zd转换说明。