1. 简述C语言关键字volatile的含义:

volatile的含义是向编译器指明变量的内容可能会由于其他程序的修改而变化。在用volatile声明变量后，优化器在用到这个变量时必须每次都小心地重新读取这个变量的值，而不是使用保存在寄存器里的备份。万一变量所存入的地方的内容被改变而变量在寄存器的变量没有变，这样声明可以保证引用这个变量的值是一开始存入的值。

1. 简述C语言关键字static的含义:

static出现在不同的地方含义是不同的。如果是在函数之外使用，表示该对像在此文件中是全局可访问的，在文件之外是不可访问的。如果出现在函数内部，则表示该变量不是自动变量，它是一个可以初始化的变量(如果不进行显式初始化，默认值是0)，改变了它的值以后，函数结束时它的值也会保留。这样的变量的寄存其实可以看成是一个全局变量，放在和全局变量一样的内存地区。

1. 什么是格式化输出？有那些函数可以实现格式化输出？

在进行输出的时候，其实都是一个字符串，但是这个字符串被输出时候有可能当成整数输出，也有可能是字符串输出，还有可能是其他的数据类型进行输出。对于输出的变量，还可能有输出的小数位不同，输出的位数不同，对齐方式不同，输出的方式不同这个计算机自己是不知道规则的，需要按照计算机能读懂的的规则告诉计算机，按照计算机能读懂的的规则告诉计算机如何输出的过程就是格式化输出。

1. &和&&的区别是什么？

&是按位与，意思是将两个表达式的值按二进制位展开，对应的位（对应的bit）按值进行“与”运算，结果保留在该位上。即（都为1结果才为1，有0结果就是0）

&&是逻辑与，意思是计算两个表达式的逻辑真值，若两个表达式都为真（非0）结果为真（非0），若有任意一个表达式为假（0）结果为假（0）。&&运算具有短路性，若&&运算符左边的表达式结果为假，则不会再去计算右边表达式的值

。

1. C语言中字符串是怎样存储的？

要么使用字符数组来存储：将字符串数据的每一个字符存储到字符数组中，并追加一个’\0’代表存储结束。要么使用字符指针来存储字符串数据：直接将一个字符串数据初始化给一个字符指针。

当它们都是局部变量的时候：

字符数组是申请在栈区，字符串的每一个字符存储在这个字符数组的每一个元素中；  
指针变量是声明在栈区的，字符串数据是以字符数组的形式存储在常量区的，指针变量中存储的是字符串在常量区的地址。

当它们作为全局变量的时候：

字符数组是存储在常量区的，字符串的每一个字符存储在这个字符数组的每一个元素中；  
指针变量也是存储在常量区的，字符串数据是以字符数组的形式存储在常量区的，指针变量中存储的是字符串在常量区的地址。

1. 如何限制一个变量只能在本文件中使用？

若该变量是全局变量，则在定义该变量时在前面加上限定词static，声明该变量是静态全局变量，只能在文件内部访问。若定义的变量是在函数内定义的局部变量，则其也只能在函数内访问，不能在函数外访问，更不能在文件外部访问。

1. 数组作为参数传入指针，传入的是数组的值还是指针？

传入的是指针。数组作为函数参数传递，值传递就是指针地址的传递，数组作为参数传递给函数的只是数组首元素的地址，函数在需要用到后面元素时再按照这个地址和数组下标去查找。作为函数形参的数组和指针可以互换，这种转换仅限于函数形参的声明。

1. 写一个宏定义（如下）

#define MIN(x,y) (((x)<=(y))?x:y)

1. （如下）

unsigned int bit\_turning­­\_1(unsigned int a)

{

unsigned int b = 1;

b = b<<4;

return a&(~b);

}

unsigned int bit\_turning\_2(unsigned int a)

{

unsigned int b = 1;

b = b<<4;

return a|b;

}