堆区、栈区和静态区是c语言中内存分配的三种区域。

栈区: 栈区是在程序运行过程中自动分配的内存区域，用于存储局部变量、函数调用信息等。在C语言中，局部变量会自动存储在栈区。栈区的大小通常由编译器决足，但可以通过编译器选项进行设置。栈内存分配运算内置于CPU的指令集，效率很高，但是分配的内存量有限

堆区: 堆区是在程序运行过程中动态分配的内存区域，用于存储动态分配的对象。在c语言中，可以使用mallo、calloc、realloc等函数来分配堆内存.编译器不会负责它们的释放工作，需要用程序区释放。

静态区: 静态区是在程序编译时分配的内存区域，用于存储全局变量、静态变量等。在C语言中，使用static关键字声明的变量存储在静态区。静态区的大小通常由编译器决定，但也可以通过编译器选项进行设置。初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。程序结束后，由系统释放。

static的作用

1、修饰局部变量：在修饰局部变量的时候，static修饰的静态局部变量只执行一次，而且延长了局部变量的生命周期，直到程序运行结束以后才释放。

2、修饰全局变量：static修饰全局变量的时候，这个全局变量只能在本文件中访问，不能在其它文件中访问，即便是extern外部声明也不可以。

静态全局变量的作用域仅限于本文件，不会被其他文件所访问、修改。当其他文件中使用相同名字的变量时，不会发生冲突。

3、修饰函数：static修饰一个函数，则这个函数只能在本文件中调用，不能被其他文件调用。

static关键字修饰函数或是全局变量时，起到的都是限定作用域的作用。

volatile的作用

volatile 定义变量后：假设有读、写两条语句，依次对同一个 volatile 变量进行操作，那么后一条的读操作不会直接使用前一条的写操作对应的 volatile 变量的寄存器内容，而是重新从内存中读取该 volatile 变量的值。

volatile 会告诉编译器，不要对 volatile 声明的变量进行各种激进的优化（甚至将变量直接消除），从而保证程序员写在代码中的指令一定会被执行。

volatile 的特性是“顺序执行”，即能够保证 volatile 变量间的顺序性不会被编译器进行乱序优化。