1. typedef使用
   1. 起别名 - 简化struct关键字
   2. 区分数据类型
   3. 提高代码移植性
2. void使用
   1. 不可以利用void创建变量 无法给无类型变量分配内存
   2. 用途：限定函数返回值，函数参数
   3. void \* 万能指针 可以不通过强制类型转换就转成其他类型指针
3. sizeof用法
   1. 本质：不是一个函数，是一个操作符
   2. 返回值类型 unsigned int无符号整型
   3. 用途：可以统计数组长度
4. 变量的修改方式
   1. 直接修改
   2. 间接修改
   3. 对自定义数据类型做练习
5. 内存分区
   1. 运行前
      1. 代码区 共享 只读
      2. 数据区 存放数据：全局变量 、静态变量、常量
         1. 已初始化数据区 data
         2. 未初始化数据区 bss
   2. 运行后
      1. 栈 符合先进后出数据结构，编译器自动管理分配和释放，有限容量
      2. 堆 容量远远大于栈，不是无限。手动开辟 malloc 手动释放 free
6. 栈区
   1. 符合先进后出数据结构
   2. 注意事项：不要返回局部变量的地址，局部变量在函数执行之后就被释放了，释放的内存就没有权限取操作了，如果操作结果未知
7. 堆区
   1. 利用malloc在堆区创建数据
   2. 利用free释放堆区
   3. 注意事项：主调函数没有分配内存，被调函数需要用更高级的指针去修饰低级指针，进行分配内存
8. static 和extern 区别
   1. 特点：在运行前分配内存，程序运行结束 生命周期结束 ，在本文件内都可以使用静态变量
   2. extern 可以提高变量作用域
9. 常量
   1. const修饰的变量
      1. 全局变量
         1. 直接修改 失败 ，间接修改 语法通过，运行失败，受到常量区保护
      2. 局部变量
         1. 直接修改 失败 ， 间接修改 成功，放在栈上
   2. 字符串常量
      1. vs 将多个相同字符串常量看成一个
      2. 不可以修改字符串常量
      3. ANSI并没有制定出字符串是否可以修改的标准，根据编译器不同，可能最终结果也是不同的