1. STL概论

为了建立数据结构和算法的一套标准，并且降低他们之间的耦合关系，以提升各自的独立性、弹性、交互操作，诞生了STL（Standard LTemplate ibrary 标准模板库）。软件界一直希望建立一种可重复利用的东西，以及一种得以制造出“可重复运用的东西”的方法，从函数，类别，函数库，类别库，各种组件，从模块化设计到面向对象，为的就是复用性的提升。

* 1. STL六大组件

容器、算法、迭代器、仿函数、适配器、空间配置器

* + 1. 容器：各种数据结构，如vector、list、deque、set、map等存放数。容器是一种class template。
    2. 算法：各种常用的算法，如 sort、find、copy等。算法是一种function template、
    3. 迭代器：扮演了容器和算法之间的胶合剂，简单理解为指针。所有STL容器都附带有自己专属的迭代器。迭代器是一种将指针相关操作予以重载的class template。原生指针也是一种迭代器。
    4. 仿函数：行为类似函数，可作为算法的某些策略。仿函数一种重载了operator()的class或class template。
    5. 适配器：一种用来修饰容器或者仿函数或迭代器接口的东西。
    6. 空间配置器：负责空间的配置和管理，配置器是哟个实现了动态空间配置、空间管理、空间释放的class tempalte。

关系：容器通过空间配置器取得数据存储空间，算法通过迭代器存储容器中的内容，仿函数可以协助算法完成不同的策略的变化，适配器可以修饰仿函数。

* 1. STL优点
     1. 内建在C++编译器中，不需要安装额外内容
     2. 不需要了解具体实现内容，只要熟练运用
     3. 高复用、可移植、高性能
  2. 容器划分
     1. 序列式容器：强调值的排序，每个元素有固定的位置。如vector、Deque、List等。
     2. 关联式容器：各元素没有严格的顺序关系，如二叉树。另一个特定是：在值中选择一个值作为关键字key，这个关键字对值起到索引的作用。
  3. 算法
     1. 质变算法：拷贝、替代、删除等。
     2. 非质变算法：查找、遍历等。
  4. 迭代器 五种

输入迭代器、输出迭代器、前向迭代器、双向迭代器、随机迭代器。（最后两种用的比较多）。

----03