1. STL概论
   1. 六大组件
      1. 容器、算法、迭代器、仿函数、适配器（配接器）、空间配置器
   2. 容器 Container
      1. 序列式容器
      2. 关联式容器 有个key起到索引作用
   3. 算法 algorithm
      1. 质变算法
      2. 非质变算法
   4. 迭代器 iterator
      1. 双向
      2. 随机访问 （功能最强 ，可以支持跳跃式的访问）
   5. 优点
      1. 不需要安装东西，直接使用
      2. 高复用性 、高移植、高性能
      3. 数据和算法分离，利用迭代器进行沟通
      4. 程序员可以不用思考 STL 具体的实现过程，只要能够熟练使用 STL 就 OK 了。这样他们就可以把精力放在程序开发的别的方面
2. 容器算法迭代器初识
   1. vector<类型> v 容器
   2. 尾插 v.push\_back()
   3. 起始迭代器 v.begin() 指向容器中第一个元素
   4. 结束迭代器 v.end() 指向容器中最后一个元素下一个位置
   5. for\_each遍历 引入头文件 algorithm
   6. 练习 内置数据类型、 自定义数据类型、 容器嵌套容器
3. string容器
   1. 构造、赋值
   2. 字符存取 [] at 区别
      1. at 访问越界 抛出 out\_of\_range异常
      2. [] 直接挂掉
   3. 拼接、查找、替换
      1. += append 拼接
      2. 查找 find 查不到 返回 -1 rfind 从右往左查
      3. 替换 replace
   4. 比较 compare
      1. 项目 = 1 字符串1 大 > 0 字符串1 小 <0
   5. 子串
      1. string substr(int pos = 0, int n = npos) const;//返回由pos开始的n个字符组成的字符串
      2. 练习 截取 email中用户名
      3. 练习2 将网址中单词截取到vector容器中
   6. 插入 insert
   7. 删除 erase
   8. char \* 和 string 转换
      1. char \* 转string 调用string有参构造 string str( “char \*”)
      2. string 转 char\* .c\_str() const char \*
      3. 编译器 可以将 char \* 隐式类型转换为 string，反之不可以
   9. 小写转大写 和 大写转小写
      1. 转大写 toupper
      2. 转小写 tolower
4. vector容器
   1. 与数组类型，但是是动态数组
   2. 动态分配内存 并不是原有空间下分配，而是找一个新空间，将原有数据拷贝到新空间下，然后释放掉原有空间
   3. 构造、赋值
   4. 交换 swap
   5. size 容器的大小
   6. capacity 容器的容量
   7. empty 容器是否为空
   8. resize 重新指定容器长度
      1. 如果比原来长了，默认用0填充，可以用第二个参数代替默认值
      2. 如果比原来短了，超出的部分元素就被删除掉了
   9. reserve 预留空间，但是不初始化
   10. at、[] 对元素存取，at越界抛出异常 ，[]直接挂掉
   11. 第一个元素 front
   12. 最后一个元素 back
   13. 插入 insert（迭代器）
   14. 删除 erase(迭代器 )
   15. 清空 clear
   16. 尾插 push\_back
   17. 尾删 pop\_back
   18. 小练习：
       1. 巧用swap来收缩内存
       2. 巧用reserve预留空间
   19. 逆序遍历 迭代器 reverse\_iterator
   20. 如果判断一个容器的迭代器是否支持随机访问
5. deque容器
   1. 双端数组 没有容量 内部用中控器控制连续分段空间
   2. 可以对头部进行插入删除
   3. 和vector的测试接口差不多
   4. 不同 头部插入 push\_front 头部 删除 pop\_front
6. 作业布置