1. 函数对象
   1. 1、函数对象通常不定义构造函数和析构函数，所以在构造和析构时不会发生任何问题，避免了函数调用的运行时问题。
   2. 2、函数对象超出普通函数的概念，函数对象可以有自己的状态
   3. 3、函数对象可内联编译，性能好。用函数指针几乎不可能
   4. 4、模版函数对象使函数对象具有通用性，这也是它的优势之一
   5. 可以作为函数参数 传递
2. 谓词
   1. 普通函数 或者仿函数 返回值 是 bool类型，这样的函数或者仿函数称为谓词
   2. 一元谓词
      1. 找容器中第一个大于20的数字 find\_if
   3. 二元谓词
      1. 对容器进行排序 sort
3. 内建函数对象
   1. template<class T> T negate<T>//取反仿函数
   2. template<class T> T plus<T>//加法仿函数
   3. template<class T> bool greater<T>//大于
   4. 头文件 #include < functional >
4. 适配器
   1. 函数对象适配器
      1. bind2nd 或者 bind1st 将两个参数进行绑定
         1. bind2nd 绑定顺序是一致
      2. 继承 binary\_function<类型1 ，类型2 ，返回值类型>
      3. 加 const
   2. 取反适配器
      1. 一元取反 not1
         1. 继承 unary\_function<类型1 ，返回值类型>
         2. 加const
      2. 二元取反 not2
   3. 函数指针适配器
      1. ptr\_fun 将函数指针 适配为函数对象
   4. 成员函数适配器
      1. mem\_fun\_ref 存放对象本体
      2. mem\_fun 存放对象指针
5. 常用遍历算法
   1. **for\_each** 主要功能：遍历
   2. 有返回值
   3. 可以绑定参数 进行输出 （适配器）
   4. transform 搬运 ，目标容器 要有容量
6. 常用查找算法
   1. **find 按值查找**
   2. **find\_if** 按条件进行查找
   3. adjacent\_find算法 查找**相邻重复**元素
   4. binary\_search算法 二分查找法
      1. 有序序列
   5. **count 按值统计**
   6. **count\_if 按条件统计**
7. 常用排序算法
   1. merge 合并
      1. 将两个有序的容器 合并到另一个容器中，合并后依然是有序
      2. 目标容器要有容量
   2. **sort** 
      1. 排序
   3. **random\_shuffle 洗牌**
   4. reverse 反转
8. 常用拷贝和替换算法
   1. copy 复制 目标容器有容量
   2. replace 替换 按值替换
   3. replace \_if 按条件替换
   4. swap 交换
9. 常用算数生成算法
   1. 头文件 #include < numeric>
   2. accumulate 累加算法
   3. fill 填充
10. 常用集合算法
    1. set\_intersection算法 求两个set集合的交集
    2. set\_union算法 求两个set集合的并集
    3. set\_difference算法 求两个set集合的差集