Logitech HW02

1. Write a class “TRADE_DATE” to store a period of time with the following member variables:
   1.1. int ID;
   1.2. tuple <int, int, int> begin_date
   1.3. tuple <int, int, int> end_date
   1.4. int druation; //the number of days in this period

2. Write a random TRADE_DATE generator (by std::random_device), which can generate 100,000 TRADE_DATE objects and store in a container
   2.1. note that the duration should be positive value between 1 to 10 days for each object. And all the year is between 1900 to 2000 A.D.

3. Write a class “intersection_finder”
   - print out all the TRADE_DATE objects that have overlaps
     - first, sort all the objects according to the begin_date with the std::sort, with the following predicates (write and test all of them)
       - an operator < in the class
       - an additional functor
       - a lambda function
     - second, use std::lower_bound/std::upper_bound to probe the possible intersection each object one after one:
       - For example:
         - auto low=std::lower_bound (v.begin(), v.end(), aaa);
         - auto up= std::upper_bound (v.begin(), v.end(), bbb);
       - check all objects in the range between “low” and “up”, and calculate the length of the overlapping days
     - The result should look like:
       - TRADE_DATA ID: X overlaps with TRADE_DATE ID: Y by Z days.
         - print out the date of X and Y to validate by eyeball check

4. Use std::bind to create std::function that can print out all intersection of a TRADE_DATE object in the vector
   4.1. std::function < void(TRADE_DATE) > new_function = std::bind (intersection_finder, …);