

## C++ Homework 09

1. Write a POD 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 duration; //the number of days in this period
2. Write a random TRADE\_DATE generator, 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 template 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
      - an 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(), a);
        - auto up= std::upper\_bound (v.begin(), v.end(), a);
      - 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 eye examination
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 = bind (intersection\_finder, ...);