Homework 03

1. Install and setup google unit test framework (you can find a tutorial here: https://github.com/google/googletest/blob/master/googletest/docs/Primer.md) in your environment.

   Try to make the following test work with your own Complex class:

```
TEST_ComplexMultiplication(ComplexMultiUnitTests, testMultiply)
{

    Complex x1(10,9); //10+9i
    Complex x2(5,4);  //5+4i
    Complex y(14,85) //14+85i
    EXPECT_EQ(x1*x2, y);
}
```

2. Please email me (juihunghung@gmail.com or jhh@cs.nctu.edu.tw) a pattern sheet (see http://wiki.c2.com/?AlexandrianForm) of your regular coding tasks.

3. Read the std::Algorithm, iterator, and container-related sections in http://en.cppreference.com/w/

4. Performance comparison of different containers, try to pass the following tests:

```
TEST_speed_backinserter(backinserter_test1, test_vector)
{
   BackInserter <int, std::vector>  data;//noted the use of a
template template parameter

   std::random_device rd;
   std::mt19937 gen(rd());
   std::uniform_int_distribution<> dis(0,65535);

   std::chrono::steady_clock::time_point start   =
std::chrono::steady_clock::now();

   for (int i=0;i<100000;++i)
   {
         data.insert( dis(gen) );
   }

   std::chrono::steady_clock::time_point end =
std::chrono::steady_clock::now();

   std::cout <<"Time elapsed: "<<
std::chrono::duration_cast<std::chrono::microseconds>(end -
start).count()<<"us\n";

}


TEST_speed_backinserter(backinserter_test2, test_XXX)
{…} // please try queue, deque, list, and set as the container
```