Due: 11:59 p.m., 25 May 2016

Problem. Write the following tests and add them to the file "test.h" in the zipped file hw8.zip. Name your tests like this: The unit test for problem 1 should have a name like TEST (p1, hw8), problem 2 TEST(p2, hw8), etc. Your solution will be graded on how closely you implement the unit test according to the requirements described in the problem. Also, your tests should be passing instead of failing.

1. (10 points) Put the 10 integers {-3, 9, 3, 0, 5, -7, 3, 6, -6, 1} in a container of the type std::vector. Use the function std::find to check that -7 is in the container and that 8 is not in there. **Information about std::find can be found here.**
2. (10 points) Repeat problem 1. with a container of the type std::list.
3. (10 points) Put the 10 integers {-3, 9, 3, 0, 5, -7, 3, 6, -6, 1} in a C array, sort the integers in the non-decreasing order with std::sort, and check that the numbers in the array is sorted. **Information about std::sort can be found here.**
4. (10 points)Repeat problem 3. with a container of the type std::vector.
5. (10 points) Repeat problem 4, but sort the numbers in the non-increasing order. You may need to write an additional function or use another function from the std library.
6. (10 points) Put the 10 integers {-3, 9, 3, 0, 5, -7, 3, 6, -6, 1} in a C array. Given the function square as follows:

   ```
   void square (int & x) {x = x*x;}
   ```

   Use the std::for_each to apply the square function to the integers in the array. Check that the numbers are correctly squared. **Information about std::for_each can be found here.**
7. (10 points) Repeat problem 6. with a container of the type std::vector.
8. (30 points) This problem involves two tests and a template function to be used by two tests specified in a. and b. below. The template function has the name minusOne, which subtracts 1 from the number that is passed in. Implement the template function minusOne. Then write two tests:
   a. (15 points) Put the 10 integers {-3, 9, 3, 0, 5, -7, 3, 6, -6, 1} in a container of the type std::vector. Use the function std::for_each to apply the function minusOne<int> to the integers in the container. Check that each number has 1 subtracted from it correctly.
   b. (15 points) Put the 10 doubles {-3.0, 9.0, 3.0, 0.0, 5.0, -7.0, 3.0, 6.0, -6.0, 1.0} in a container of the type std::list. Use the function std::for_each to apply the function minusOne<double> to the doubles in the container. Check that each number has 1 subtracted from it correctly.
9. (Bonus, 50 points) A template function called find_all is needed that returns a

std::vector of numbers that make the template function isPositive return true. For example,

```
int a[10] = {-3, 9, 3, 0, 5, -7, 3, 6, -6, 1};
std::vector<int> b=find_all(a,a+10,0,isPositive<int>);
```

b should contain {9,3,0,5,3,6,1}.

a. (30 points) Write the function find_all, the function isPositive, and convert the example into a test that passes.

b. (20 points) Repeat problem 9a with 10 doubles {-3.0, 9.0, 3.0, 0.0, 5.0, -7.0, 3.0, 6.0, -6.0, 1.0} stored in a std::list. Note that the call to find_all becomes:

```
std::vector<double> b=find_all(a.begin(),a.end(),0.0,isPositive<double>);
```