Homework #1 – STL Practice

(This is a good reference to the C++ standard library: http://www.cplusplus.com/reference/).

You may not use any looping constructs (e.g. `for`, `while`, `do`) in this assignment.

1. **(3 points)** Write a unit test demonstrating **deque** & **algorithm** functionality. This test should take the following steps:
    a. Use the `generate_n` algorithm to populate a `deque` with the values 1 through 10 (you may not call `insert`, `push_back`, or `push_front`). *Hint: Pass a `back_inserter` to the call to `generate_n` to cause the values to be pushed onto the back of the `deque`.*
    b. Use the `accumulate` algorithm to compute the sum of the values in the `deque`.
    c. Verify that the sum is 55.
2. **(3 points)** Write a unit test demonstrating **string** & **algorithm** functionality. This test should take the following steps:
    a. Create a `string` containing the lowercase letters of the alphabet in ascending order.
    b. Use the `transform` algorithm to convert the `string` to uppercase.
    c. Verify that the `string` contains the uppercase letters of the alphabet in ascending order.
3. **(3 points)** Write a unit test demonstrating **vector** & **algorithm** functionality. This test should take the following steps:
    a. Create an array literal containing the values 10 through 1.
    b. Create a `vector` initialized to contain the values in the array by using the `vector's` iterator range constructor.
    c. Use the `partition` algorithm to place the even numbers in the lower half of the `vector` and the odd numbers in the upper half of the `vector`.
    d. Use the `sort` algorithm to sort each partition.
    e. Use the `copy` algorithm to copy the `vector`'s elements to an `ostring_stream`. *Hint: Pass an `ostream_iterator` bound to the `ostring_stream` as the destination parameter to the `copy` algorithm; this will cause `copy` to write the copied values to the underlying stream.*
    f. Verify that the `ostring_stream` contents are "24681013579"
4. **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, follows a consistent style, and is ANSI-compliant.

Place all source code and a screen capture of the output produced by your program in a single PDF document. Submit this document.