Assignment 8

1. Study Nguyen, Chapter 4.2 subsections Implementation of HeapSorter, Implementation of SelectSorter and InsertSorter, and Section 4.3.

2. Do Nguyen Exercise 4–7(a) to rewrite the siftDown() function in Heapifier.hpp without recursion. Heapifier.hpp is in the ASorter project. Note the optimization requirements in the problem statement.

   Hint: Here are two lines of code to get you started. Terminate your while loop on !done where done is a boolean variable, and set done accordingly in the body of your loop. Do not use the C++ break statement.

   ```cpp
   int child = 2 * i - lo + 1; // Index of left child.
   bool done = hi < child;
   ```

   First, test your program with the SortInt project, which sorts integers.

   Then, test your program with the SortCompAsgn project. In addition to SortCompAsgn, you will need to install the following projects on which SortCompAsgn depends:

   CAMetrics  
   Counter

   Build only the SortCompAsgn project. Do not build CAMetrics or Counter as neither of them have a main program. You will need to include CAMetrics and Counter along with ASorter and the usual projects in your list of include files for SortCompAsgn.

   Test your program with data from the ten files d0200, d0400, d0800, d1600, d2400, d3200, d4000, d4800, d5600, d6400, which contain 200, 400, 800, 1600, 2400, 3200, 4000, 4800, 5600, and 6400 random real values respectively.

   Hand in the following three files:

   ```cpp
   MergeSorter.hpp  
   Heapifier.hpp  
   InsertSorter.hpp
   ```

   Fix any bugs that may have been in MergeSorter.hpp from Assignment 6 and InsertSorter.hpp from Assignment 7. This code will be the basis for your data collection for the paper in Assignment 9.