The purpose for this experiment is to provide you experience measuring, analyzing, and discussing computer performance. We are analyzing and comparing the performance of the following two quadratic complexity sorting algorithms: selection sort vs insertion sort.

See [Sorting Algorithms Animations | Toptal®](https://www.toptal.com/developers/sorting-algorithms)

You will also need to create appropriate test data, usually via randomization.  
Some tests will repeat sorts on the same data (same random seed) in order to estimate the randomness of system overhead, whereas the other tests will be randomizing the data for each run.

Measure the following metrics (try to isolate each):

* Wall clock time
* CPU time
* In the loop body:
  + Count and rate of swap operations
  + Count and rate of comparison operations
* Profile
* Relative (ratio) to your programming language’s sorting library function’s performance

Investigate the effect of:

* Profiling latency
* Varying the Compiler optimization levels
* Partially sorted data (varying the proportion of adjacent elements that are initially in the correct order (random data should be at ~50%)
* Varying the ratio of unique elements to total elements (the effect of duplicates)
* Integer vs floating point data (of same count of bytes per element)

Statistics:

* Range of values
* (appropriate) mean & median
* Standard deviation