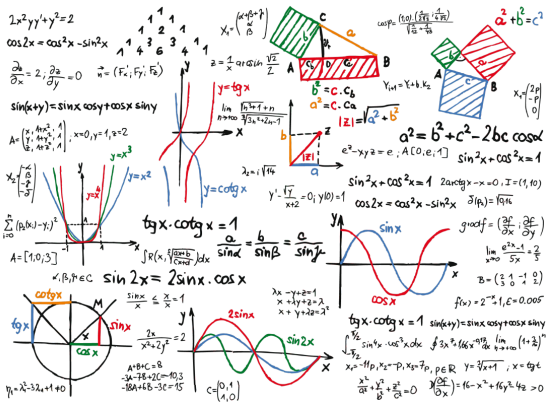

B-1- Mathematics

B-MAT-500

301dannon

Benchmark Sorting

v1.3

301dannon

Benchmark Sorting

binary name:	301dannon
repository name:	301dannon
repository rights:	ramassage-tek
language:	C, C++, perl 5, python 3 (≥ 3.5), ruby 2 (≥ 2.2), php 5.6, bash 4
group size:	1 to 2
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

You've been hired by Dannon to sort the information from their partners' (Google, Facebook, Deezer) databases, in order to sort and process it. There are a few thousand petabytes of data, and it is crucial to execute the most optimal sorting possible.

Therefore, you are going to perform a benchmark of different sorting algorithms by comparing their execution speed, or rather the number of effectuated operations (in order to be independent of the machine and its power.)

The algorithms involved are as follows:
- selection sort
- insertion sort
- bubble sort
- quicksort
- merge sort

There are numerous possible operations: variable declaration, variable affectation, function call, accessing variables, calculation, comparison, test etc. You will only take into account the comparisons that are meaningful in sorting.

For selection sort, insertion sort and bubble sort, we will start with the beginning of the list.
For quicksort, we will always take the list's first element as a pivot.

You could also add some nice extra features as bonus:
- add other algorithms
- benchmark the algorithms' optimizations in order to see which ones are truly efficient
- produce statistics, draw curves for the number of operations according to the size of the data

Don't optimize your algorithms or you will skew the result.

Terminal

```
~/B-MAT-500> ./301dannon -h
USAGE
          ./301dannon file

DESCRIPTION
        file       file that contains the numbers to be sorted, seperated by spaces
```

The sorting functions are obviously unauthorized, regardless of the language. You must code the sorts yourself.

Examples

Terminal

```
~/B-MAT-500> cat list
3.3 5 9.89 -6
~/B-MAT-500> ./301dannon list
4 elements
select sort:  6 comparisons
insertion sort:  4 comparisons
bubble sort:  6 comparisons
fusion sort:  5 comparisons
quicksort:  4 comparisons
```

Terminal

```
~/B-MAT-500> cat list2
-564 1340 42 129 858 1491 1508 246 -1281 655 1506 306 290 -768 116 765 -48 -512 2598 42 2339
~/B-MAT-500> ./301dannon list2
21 elements
select sort:  210 comparisons
insertion sort:  125 comparisons
bubble sort:  210 comparisons
fusion sort:  67 comparisons
quicksort:  80 comparisons
```