1) Use the files `s_100000.txt` and `s_1000000.txt`. These files contain lists of randomly ordered integers. For example, the first few lines of `s_100000.txt` are shown below

```
100000
/**/
61055
60294
60593
13850
  .
  .
  .
```

which file contains the integers `0` through `99999` (inclusive) ordered randomly. The file `s_1000000.txt` contains the integers `0` through `999999` (inclusive) ordered randomly.

2) Use the programs `RecursiveMergeSort.java` and `QuickSortInt.java`, both of which need `package IO`. The program `RecursiveMergeSort`, which we have looked at previously, sorts a list of numbers with the recursive merge sort algorithm. The program `QuickSortInt` sorts a list of numbers with the quick sort algorithm, which algorithm is also recursive. As you will see, both algorithms run in about the same time, *i.e.*, of $O\ (n \log n)$.

3) Compile both `RecursiveMergeSort.java` and `QuickSortInt.java`. Both of these programs can read in the files `s_100000.txt` and `s_1000000.txt`, perform the sort, and they will print, to the console, the time *t* (in ms) that was required to perform the sort. Run the programs and fill in the table below.

n	*t* (ms) merge sort	*t* (ms) quick sort
100 000		
1 000 000		