

ComS 362 HW 1

1. The users may want the randomly generated numbers to be written to a file.

```
// perform quick sort
fileData = makeCopy(fileMaster);
startTime = getTime();
qSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "quick Sort",
    fileData.length, endTime - startTime);
//Add file-writing module here
} else {...
```

This code can be found at lines 52-59. For this change, you wouldn't necessarily want to change any code, but you would need to add on code that could take the variable fileMaster and copy its data to a file. The user would most likely want a way to name the file, which would be easy enough by reusing getFileName().

2. The users may want to change the time format that is printed out (i.e. nano, micro, milli, second).

```
do {
// input data source
prompt("Data source? (0 = file, 1 = generated)");
//Add additional prompt here
...

// perform insertion sort
fileData = makeCopy(fileMaster);
startTime = getTime();
insertionSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "insertion Sort",
    fileData.length, endTime - startTime);

// perform selection sort
fileData = makeCopy(fileMaster);
startTime = getTime();
selectionSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "selection Sort",
    fileData.length, endTime - startTime);
...
}
```

This code can be found at lines 13-18 and 31-43. For this change, you need add an addition prompt for the user to select the format they want and a module to process their

request. You would also need to add a module to print the correct format and replace every print statement with it.

3. The users may want the prompts to be displayed in multiple languages.

```
do {  
    //Add additional prompt here  
  
    // input data source  
    prompt("Data source? (0 = file, 1 = generated)");  
  
    ...  
  
    // perform insertion sort  
    fileData = makeCopy(fileMaster);  
    startTime = getTime();  
    insertionSort(fileData);  
    endTime = getTime();  
    System.out.format("%20.20s %10d %10d %n", "insertion Sort",  
        fileData.length, endTime - startTime);  
  
    // perform selection sort  
    fileData = makeCopy(fileMaster);  
    startTime = getTime();  
    selectionSort(fileData);  
    endTime = getTime();  
    System.out.format("%20.20s %10d %10d %n", "selection Sort",  
        fileData.length, endTime - startTime);  
  
    ...  
}
```

This code can be found at lines 13-18 and 31-43. For this change, you would also need an additional prompt to select the language and a module to process the selected language. For every print statement, you would need to replace it with a module to print the selected language.

4. The user may want to switch the order that the sorting algorithms are ran in.

```
// perform insertion sort  
fileData = makeCopy(fileMaster);  
startTime = getTime();  
insertionSort(fileData);  
endTime = getTime();  
System.out.format("%20.20s %10d %10d %n", "insertion Sort", fileData.length, endTime -  
    startTime);  
  
// perform selection sort  
fileData = makeCopy(fileMaster);  
startTime = getTime();  
selectionSort(fileData);  
endTime = getTime();
```

```

System.out.format("%20.20s %10d %10d %n", "selection Sort", fileData.length, endTime -
startTime);

// perform merge sort
fileData = makeCopy(fileMaster);
startTime = getTime();
mergeSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "merge Sort", fileData.length, endTime -
startTime);

// perform quick sort
fileData = makeCopy(fileMaster);
startTime = getTime();
qSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "quick Sort", fileData.length, endTime -
startTime);

```

All the highlighted code would need to be put into separate modules so they can be called in the user's order.

5. The user may want the output to be displayed as a histogram instead of numbers.

```

// perform insertion sort
fileData = makeCopy(fileMaster);
startTime = getTime();
insertionSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "insertion Sort", fileData.length, endTime -
startTime);

// perform selection sort
fileData = makeCopy(fileMaster);
startTime = getTime();
selectionSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "selection Sort", fileData.length, endTime -
startTime);

// perform merge sort
fileData = makeCopy(fileMaster);
startTime = getTime();
mergeSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "merge Sort", fileData.length, endTime -
startTime);

// perform quick sort
fileData = makeCopy(fileMaster);
startTime = getTime();
qSort(fileData);
endTime = getTime();
System.out.format("%20.20s %10d %10d %n", "quick Sort", fileData.length, endTime -
startTime);

```

The highlighted print statements will need to be replaced with modules that take the input data from each algorithm and use it to print out a histogram in the terminal.