CS 1400-03 Introduction to Programming and Problem Solving
Coding Practice #8
(Due: 11:59 PM, Friday, 4/9/2021)

Except Coding Practice #1, I will not grade your coding practice submissions. Instead, they will be treated as participation points. On blackboard, you will receive full points as long as you work on the exercises, which don't necessary mean they are all correct. Please check your own programs carefully and make sure they do generate the desired output.

Objectives:

- Be able to write complete Java programs with
  - arrays, partially filled arrays, methods, files
  - sequential search, selection sort, binary search
- Be able to test and debug a program

Change your working directory to `cs1400/codingPractice` for this assignment.

Task #1 Lottery

Write a program that simulates a lottery. Your program should have an array of five integers named `lotteryNumbers`. Use the `Random` class to generate a random number in the range of 0 through 9 for each element in the array. Your program should either read in five integers that represent a person's lottery picks or let the computer to select numbers for the user. The program should then display the number of digits that match the lottery numbers.

Write the following methods in the program:

- `Public static int[] generateLotteryNumber()` – This method should use the `Random` class to generate a random number in the range of 0 through 9 for each element in the array. Return the array of randomly generated numbers.
- `Public static int compareNumbers(int[] lottery, int[] picks)` – This method is to compare the corresponding elements in the two arrays and return the number of digits that match.

Here are two sample interactions. The user's input is shown in bold.

```
fcsang@fluffy ~/cs1400/codingPractice $ java Lottery
welcome to play Lottery ...
1. enter your own lottery number
2. get a computer-generated lottery number
your choice: 2

The Lottery Number is:
        0 4 2 4 5
Your Number is:
        9 0 2 4 7
There are 2 matching digits.

fcsang@fluffy ~/cs1400/codingPractice $ java Lottery
welcome to play Lottery ...
1. enter your own lottery number
2. get a computer-generated lottery number
```

```
your choice: 1
enter 5 single-digit numbers, separated by space: 3 4 5 6 7

The Lottery Number is:
        2 3 8 2 9
Your Number is:
        3 4 5 6 7
There are 0 matching digits.
```

Task #2 Name Search

Copy two files from Bb where `BoyNames.txt` contains a list of 200 most popular names given to boys born in the United States for the years 2000 through 2009 while `GirlNames.txt` contains a list of 200 most popular girl names.

Write a program that reads the contents of the two files into two separate arrays. The user should be able to enter a boy's name or a girl's name, and the application will display messages indicating whether the name was among the most popular.

Write the following methods in the program:

- `Public static void getNamesFromFile(String[] array, String filename)` – This method fills an array of strings with the values read from a file.
- `Public static int sequentialSearch(String[] array, String value)` – This method searches an array for a value and returns the subscript of the value if found in the array, otherwise returns -1.
- `Public static void displaySearchResult(String input, String[] array1, String[] array2)` – This method calls the `sequentialSearch()` method twice in order to determine if the name input by the user is located in the list of popular boy and girl names. Once the search is complete, the result is displayed to the user. The parameter `array1` is the array containing the most popular boy's names and `array2` is the array containing the most popular girl's names.

Here are sample interactions. The user's input is shown in bold.

```
fcsang@garrison ~/cs1400/codingPractice $ java NameSearch
Popular Name Search

Enter a name (blank line to stop): Daisy
Daisy is a popular girl's name.

Enter a name (blank line to stop): Richard
Richard is a popular boy's name.

Enter a name (blank line to stop): Kristine
Kristine is not a popular name.

Enter a name (blank line to stop): <enter>
```

Task #3 Selection Sort

Write a program that reads in the file called `data.txt` (available on Bb) that contains integers, each on a separate line. We don't know how many elements in the file. Your main method should create a large enough array of constant size 100, fill the array with integers from the input file, and use a counter variable to keep track of the actual number of items stored in the array. If `data.txt` does not exist, give an appropriate error message and terminate the program. After integers are stored in an array, your program should use Selection Sort and output the sorted data to a file called `data.out`.

Write the following methods in the program:

- `Public static void selectionSort(int[] array, int counter)` – This method should implement Selection Sort and sort the array of integers into ascending order. Note that the second parameter `counter` is to make sure that you only process the array elements that contain valid data items.
- `Public static void outputData(int[] array, int counter)` – This method will open an output file called `data.out`, print integers in `array`, 10 integers per line and use tab between integers on the same line.

Sample output in '`data.out`':

```
-200    -11     -9      -5      -1      0       2       4       8       22
28      29      34      45      61      62      63      64      65      88
99      100     200     1000
```

Task #4 Binary Search

Write a method to perform a binary search on an integer array. The method will return the position of the `target` element in an `array` or return -1 if `target` is not found.

Then, write a main method that creates and initializes a sorted array, asks the user for a target element, and locates the target in the array.

Here are sample interactions. The user's input is shown in bold.

```
fcsang@garrison ~/cs1400/codingPractice $ java BinarySearch
enter a number: 10
array = [ 15 20 25 30 35 40 ]
10 is not in the array

fcsang@garrison ~/cs1400/codingPractice $ java BinarySearch
enter a number: 15
array = [ 15 20 25 30 35 40 ]
15 is at position 0

fcsang@garrison ~/cs1400/codingPractice $ java BinarySearch
enter a number: 20
array = [ 15 20 25 30 35 40 ]
20 is at position 1
```

Submission:
Generate a `script` file `practice8.txt` with appropriate time stamps and the following steps visible:

1) a `pwd` to show the current working directory
2) a `ls -l` to show in long format the files in your `cs1400/codingPractice` directory
3) display `Lottery.java`
4) compile `Lottery.java`
5) run `Lottery`
6) display `NameSearch.java`
7) compile `NameSearch.java`
8) run `NameSearch`
9) display `SelectionSort.java`
10) compile `SelectionSort.java`
11) run `SelectionSort`
12) display `BinarySearch.java`
13) compile `BinarySearch.java`
14) run `BinarySearch`

Submit the script file `practice8.txt` on Bb, under the Coding Practice Folder, Practice #8 link.