Java Lab1: 10 points

In this lab, you will learn to use: Console I/O, Strings, Arrays, and Loops

Problem statement: Write a program that takes n number of names as strings, prints them out in alphabetically sorted order, and then searches for a name input by the user in the sorted strings. As shown in Fig.1 and Fig.2, the user may enter the names in upper or lower case. In other words, your program should be *case-insensitive*. If the name is not found, the output should be as shown in Fig.3. If the number of names to store as entered by user is 0 or a negative number, then the program should simply print 'GoodBye!' and exit.

Solution Design: The program has one class NameSorter() that has three methods as shown in Figure 5.

The main() method gets the program started, and takes user inputs and prints some output. It invokes the getNameInputs() and sortAndSearch() methods as needed. While main() method is fully coded for you, you need to write the code for the remaining two methods. The specifications for these two methods is provided in the NameSorter.java file as comments.

Hint: To sort the names, look up java docs to find a class named Arrays and its method sort().

Instructions:

1. Download NameSorter.java and TestNameSorter.java from Canvas
2. Create a package named lab1 and import these two files into this package
3. Fill in your code in NameSorter.java. <u>Do not change</u> any method signatures.
4. Test your NameSorter.java using TestNameSorter.java. <u>Do not change</u> TestNameSorter.java.
5. Make sure your console output also comes out as expected.
6. Write your Andrew-id and name as top line comment in your **NameSorter.java** file **<u>only</u>** and submit it on Canvas.

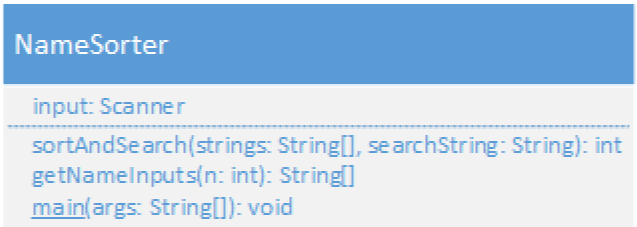
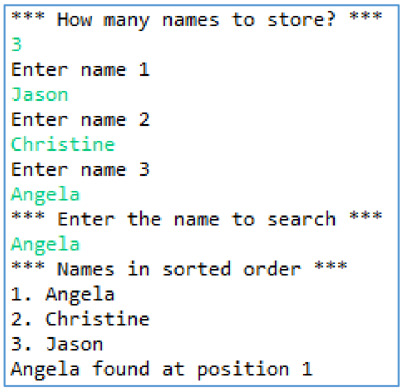
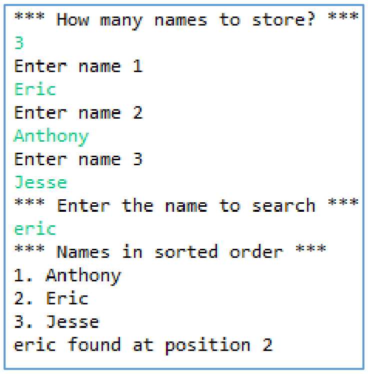

NameSorter
input: Scanner
sortAndSearch(strings: String[], searchString: String): int
getNameInputs(n: int): String[]
main(args: String[]): void

Figure 5: Class diagram



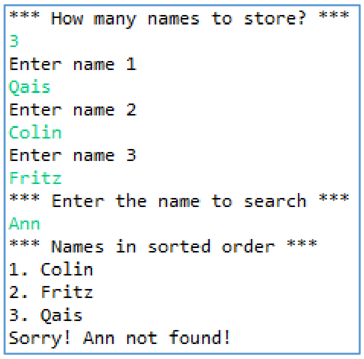
```
*** How many names to store? ***
3
Enter name 1
Jason
Enter name 2
Christine
Enter name 3
Angela
*** Enter the name to search ***
Angela
*** Names in sorted order ***
1. Angela
2. Christine
3. Jason
Angela found at position 1
```

Figure 1: n = 3; Same case search string found



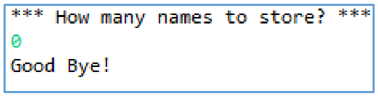
```
*** How many names to store? ***
3
Enter name 1
Eric
Enter name 2
Anthony
Enter name 3
Jesse
*** Enter the name to search ***
eric
*** Names in sorted order ***
1. Anthony
2. Eric
3. Jesse
eric found at position 2
```

Figure 2: n=3; Different case Search string found



```
*** How many names to store? ***
3
Enter name 1
Qais
Enter name 2
Colin
Enter name 3
Fritz
*** Enter the name to search ***
Ann
*** Names in sorted order ***
1. Colin
2. Fritz
3. Qais
Sorry! Ann not found!
```

Figure 3: n = 3. Search String not found



```
*** How many names to store? ***
0
Good Bye!
```

Figure 4: n = 0;

Rubric:

- 5 test-case: 5 points (1 point each)
- 4 scenarios (Fig. 1 to 4): 4 points (1 point each)
- Coding: 1 point (Correct file submitted, code well commented, coding conventions followed)