CSIS1275-04 **Assignment 3** **Due Date: Nov. 7th/2018**

You can do this assignment with **ONE** partner (NOT 35)

Write a program with the following class and functionality.

class Assign3 (save as **Assign3.java**)

main(..)
class **Assign3** will **contain the public static void main(..) method**. It will contain an array of people's names

**String names[] = {"Carol Danvers","halo JOnes", "lex LUthor", "NEal adams", "halo Jones","theo KOjak", "david starsky",
"clark KEnt", "Carol Danvers","bruce LEE", "bruce WAYne","donald trump", "NEal adams", "DONAld TRump", "david starsky", "clark KEnt", "Bruce Lee",
"bruce WAYne","xavier chuck", "donald trump","JAck Kirby","LOis LAne",
"LAdy Penelope","DIana Prince","Carol Danvers","","","","",""};**

Note that the last couple of array slots are empty.

main(..) will invoke the following methods (and please note, when you create these array methods, they should all work for **any sized** arrays.:

sortArray(….) (instance method)
Use this method to perform a bubble sort on the **array in ascending alphabetical order (i.e. in name sequence)**. This method will take **at least** an array of String type as a parameter, **HOWEVER** remember that you **cannot** sort the entire array, as the last few array slots are empty. This method will return nothing. If you were to display the array after calling this method, it will look similar to this:

```
Alphabetical Order
1------ bruce LEE
2------ Bruce Lee
3------ bruce WAYne
4------ bruce WAYne
5------ Carol Danvers
6------ Carol Danvers
7------ Carol Danvers
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ donald trump
14------ DONAld TRump
15------ donald trump
16------ halo JOnes
17------ halo Jones
18------ JAck Kirby
19------ LAdy Penelope
20------ lex LUthor
21------ LOis LAne
22------ NEal adams
23------ NEal adams
24------ theo KOjak
25------ xavier chuck
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
========================
```

(**DO NOT USE ANY of Java's own array methods**)
(HINT: **You may need to use the "compareTo" method for string comparison**)

It will sort **alphabetically**, ***irrespective*** of **case sensitivity**. When displaying, notice the **numbers** on the **left**, start at **1** and **NOT** 0.

Note also, that the empty entries are displayed with " == Empty ==" , **HOWEVER DO NOT** actually put that into the empty slots. The empty slots should be just empty strings, i.e. "".

do bin search
in the end, when remove item, add zzzz in to make sure it stays at the end

binSrch (instance method)
This method will take at **least** a **String array** and **a String** as parameters. This method will use a binary search to locate where a person's name is in the array by **returning back the index** of where it is located. Return back **"-1**" if the name **does not exist** in the string array. Please note that this method will perform a **binary search** on a String array that is sorted in ascending sequence. It is also **important to note** that this method must be able to search irrespective of case (whether it is uppercase, lowercase or mixture of both. Please see screen dialogue on last page.). **Note also that you cannot search the entire array as the last few array slots are empty**.

(**DO NOT USE ANY of Java's own array methods**)
(HINT: **You may need to use the "compareTo" method for string comparison**)

displayArray (instance method)
This method will take at **least** a String array as a parameter , and will display the content of the array, in the format indicated on **page 1**, and nothing else. (Note, the very last line, i.e. "**==========**" is ALSO from this method, however the title i.e. `Alphabetical Order` , should **come from outside** of this method).

removeNames(instance method)
This method will take at **least** a String array as a parameter. This method will **remove all occurrences** of a particular name. **Removal** in this case means that the array slots with that name are replaced with empty strings, i.e. "". Please make use of the binary search and sort methods you wrote.
(**Hint**: rather than replace the name with "", you might want to consider initially replacing it with a large string value like "zzzzzzz" and then sort it)

You can start with the following:

```
import java.io.*;
import java.util.*;
import java.text.*;

class Assign3
{
        ////////////////////////////////////////////////
        // Your methods
        ////////////////////////////////////////////////



        public static void main(String arg[])
        {

                String names[] = {"Carol Danvers","halo JOnes", "lex LUthor", "NEal adams",
"halo Jones","theo KOjak", "david starsky",
            "clark KEnt", "Carol Danvers","bruce LEE", "bruce WAYne","donald trump", "NEal
adams", "DONAld TRump", "david starsky", "clark KEnt", "Bruce Lee",
            "bruce WAYne","xavier chuck", "donald trump","JAck Kirby","LOis LAne",
                "LAdy Penelope","DIana Prince","Carol Danvers","","","","","","};



        }
}
```

Functionality and Typical Screen Dialogue:
When the program is first run, it will display the people's names in ascending **alphabetical** order and then it will ask the user to enter a name to search for (via initially a **binary** search). If located, it will display **where they are found**, as shown on **page 2**. It will then request for the next name to search for and so on.

If the user preceded the name search with a minus sign, i.e. **`-neal adams`**, this means you want to **remove ALL** occurrences of that name.

The program will exit on the user typing in "0". If the name is not found, an appropriate error message will be displayed, as shown in the examples below.

The **approach** to this assignment **MUST** be of the following:

*Use the **binary** search to locate an occurrence of the name you are searching for. However, as there are duplicate names, that name located **may not** be the first occurrence of that name. A **linear** search may be employed to go **backwards** and **forwards** to locate the **first** and all subsequent occurrences of that name, **from the point** where the first occurrence of the name was found **by the binary search**. This approach should be used **whenever you need to search for a name** (even when **removing names**, since you need to locate the name first before removal)*

You will **lose marks** (**many**) if all you did was simply employ a linear search from the start of the array to search for names.

For the given String array, you can assume that there will always be **one space between** the first and last name and that there will be NO leading or trailing spaces between the quotes.

The following are some examples:
Ex1 (on first run)

```
Alphabetical Order
1------ bruce LEE
2------ Bruce Lee
3------ bruce WAYne
4------ bruce WAYne
5------ Carol Danvers
6------ Carol Danvers
7------ Carol Danvers
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ donald trump
14------ DONAld TRump
15------ donald trump
16------ halo JOnes
17------ halo Jones
18------ JAck Kirby
19------ LAdy Penelope
20------ lex LUthor
21------ LOis LAne
22------ NEal adams
23------ NEal adams
24------ theo KOjak
25------ xavier chuck
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
=======================
Please enter name to search or remove(-)(0 to Exit):
```

The `Please enter name to search or remove(-)(0 to Exit):` is also from main(..).

Ex2

```
Alphabetical Order
1------ bruce LEE
2------ Bruce Lee
3------ bruce WAYne
4------ bruce WAYne
5------ Carol Danvers
6------ Carol Danvers
7------ Carol Danvers
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ donald trump
14------ DONAld TRump
15------ donald trump
16------ halo JOnes
17------ halo Jones
18------ JAck Kirby
19------ LAdy Penelope
20------ lex LUthor
21------ LOis LAne
22------ NEal adams
23------ NEal adams
24------ theo KOjak
25------ xavier chuck
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
========================
Please enter name to search or remove(-)(0 to Exit):     neAL ADams
neAL ADams located in
position 22
and in position 23
========================
Please enter name to search or remove(-)(0 to Exit):
```

Above shows that the user wants to search for "**neal adams**" entered with mixed casing and with leading and trailing spaces.

Note you can assume the user will know to **enter only <u>ONE</u> embedded space** between first and last name.

When **searching** you **don't need** to redisplay the content of the array, you just need to display where it is found (**See Ex3**).

In main(..) it should continue to prompt the user to search for another name until "0" is entered, at which point the program will end.

Ex3

```
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ donald trump
14------ DONAld TRump
15------ donald trump
16------ halo JOnes
17------ halo Jones
18------ JAck Kirby
19------ LAdy Penelope
20------ lex LUthor
21------ LOis LAne
22------ NEal adams
23------ NEal adams
24------ theo KOjak
25------ xavier chuck
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
========================
Please enter name to search or remove(-)(0 to Exit):     neAL ADams
neAL ADams located in
position 22
and in position 23
========================
Please enter name to search or remove(-)(0 to Exit): donald trump
donald trump located in
position 13
and in position 14
and in position 15
========================
Please enter name to search or remove(-)(0 to Exit):
```

When **searching** you **don't need** to redisplay the content of the array, you just need to display where it is found.

Ex4

```
Please enter name to search or remove(-)(0 to Exit): -donald trump
1------ bruce LEE
2------ Bruce Lee
3------ bruce WAYne
4------ bruce WAYne
5------ Carol Danvers
6------ Carol Danvers
7------ Carol Danvers
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ halo JOnes
14------ halo Jones
15------ JAck Kirby
16------ LAdy Penelope
17------ lex LUthor
18------ LOis LAne
19------ NEal adams
20------ NEal adams
21------ theo KOjak
22------ xavier chuck
23------  == Empty ==
24------  == Empty ==
25------  == Empty ==
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
========================
Please enter name to search or remove(-)(0 to Exit):
```

Above shows what should happen if you want to remove a name. Notice the name is preceded by a minus sign. When removing names **you do need to redisplay the array after the name removals**.

Ex5

```
Please enter name to search or remove(-)(0 to Exit):   -  donald trump
1------ bruce LEE
2------ Bruce Lee
3------ bruce WAYne
4------ bruce WAYne
5------ Carol Danvers
6------ Carol Danvers
7------ Carol Danvers
8------ clark KEnt
9------ clark KEnt
10------ david starsky
11------ david starsky
12------ DIana Prince
13------ halo JOnes
14------ halo Jones
15------ JAck Kirby
16------ LAdy Penelope
17------ lex LUthor
18------ LOis LAne
19------ NEal adams
20------ NEal adams
21------ theo KOjak
22------ xavier chuck
23------  == Empty ==
24------  == Empty ==
25------  == Empty ==
26------  == Empty ==
27------  == Empty ==
28------  == Empty ==
29------  == Empty ==
30------  == Empty ==
========================
Please enter name to search or remove(-)(0 to Exit):
```

If the user entered spaces between the minus symbol and the name, that should still work.

Ex6

```
Please enter name to search or remove(-)(0 to Exit): neal     adams
neal     adams is not found
========================
Please enter name to search or remove(-)(0 to Exit): -  neal     adams
neal     adams is not found
========================
Please enter name to search or remove(-)(0 to Exit): -neal    adams
neal    adams is not found
========================
Please enter name to search or remove(-)(0 to Exit):
```

Above shows what happens if the name was **not found**. This applies also if you want to remove a name that was not found.

Incidentally, the name was not found because the user entered **more than one embedded space between the first and last name**, but that would be at user's risk.

Ex7

```
========================
Please enter name to search or remove(-)(0 to Exit): carol danvers
There are NO names left to search/remove
Please enter name to search or remove(-)(0 to Exit): -donald trump
There are NO names left to search/remove
Please enter name to search or remove(-)(0 to Exit):
```

Above shows what happens if you have removed all names from the array, but you still want to search or remove.

When testing this part you can use the following String array, to make it easier:

```
String names[] = {"Carol Danvers","Carol Danvers", "","","","","","};
```

Ex7
You assume that if the user entered **"0donald trump",**
Or any string preceded by a **"0",** this would **exit the program**.

Additional
You may want to create methods to locate the **first occurrence** and all other occurrence of the search name.

Please make any additional methods to be **instance** as well.

NOTE: You can use the Accept and Screen classes you wrote for Assignment 1or 2 for the screen clearings and accepting of data at the prompt.

Restrictions

- You **cannot** use **ArrayList** or **Vectors**.
- You **cannot** use any **StringBuilder** methods.
- You **can use String methods like "replace()", "charAt()" ,"trim()" and the comparison methods**.
- You **cannot** use any **Java's own built-in Array methods for searching or sorting**.

What to hand in:

1. Front cover page with your name (your partner's name), course, assignment number, instructor's name, section and due date. (If you did this assignment with a partner, then hand in **ONE** piece of work with **BOTH** your names on it).
2. Hardcopy of the program listing of ALL classes.
3. Zip up the files as **csis1275F2018_yourSec_Assign3YourName.zip**
4. Upload through Blackboard

(If you did this with a partner, then please hand in one piece of work with both your names on the cover page. PLEASE ENSURE THAT THIS IS YOUR OWN WORK OR MARKS WILL BE LOST OR WORSE !)