

Exercise on Arrays in Java

The program requested in this exercise will be a database of names. They will be stored in an array of strings. When a name is inserted it will be placed next to the last one. Consider the following array:

0	1	2	3	4	5	6	7	8	9
"Joe"	"Jude"	"Mary"	"Paul"						

Let us assume that the array is called "students". Note that the array's capacity is 10 elements but right now it is holding only 4. Note also that the names are sorted alphabetically.

Names can be inserted and even deleted. Suppose now that the name "Kim" needs to be inserted. Two steps will have to be executed here. In the first step "Kim" is placed next to "Paul" and it will be the last element. Then the elements are sorted. These two steps are shown in the following diagram:

Step 1: Insert "Kim" as the last element.

0	1	2	3	4	5	6	7	8	9
"Joe"	"Jude"	"Mary"	"Paul"	"Kim"					

Step 2: Sort the names

0	1	2	3	4	5	6	7	8	9
"Joe"	"Jude"	"Kim"	"Mary"	"Paul"					

Another operation is deleting a name. Suppose we want to eliminate "Jude". This will be done in these two steps:

- 1) Put the last element ("Paul" in this case) in the place of "Jude".
- 2) Sort the elements (the first 4 in this case).

In the program declare a constant to indicate the number of elements:

```
static final int NUM_OF_STUD = 10; //size of the array
```

Also declare a variable to hold the number of names present in the array:

```
int nost = 0; //number of students in array
```

Initially this variable will be equal to 0 (in the example seen above, before the deletion of "Jude", 'nost' was equal to 5).

The program should be able to do (repeatedly) the following:

- Insert a name
- Delete a name
- Display all names
- Find the number of occurrences of a name
- End

The first four options have to be implemented by means of a method each. The method to insert the names is shown below. It makes use of the method to sort names.

```
public static int insertName (String[] names, String name, int n)
//parameters:
//names: the array into which the name of student will be inserted
//name: the name of the student that will be inserted
//n: the number of students present in the array
{
    if (n<NUM_OF_STUD)
    {
        names[n] = name;
        sortNames(names, n+1);
        return 0; //return 0 means that the name has been inserted
    }
    else return -1; //return -1 means that the array is full
}
```

The method to sort names (it makes use of the bubble sort) is shown below.

```
public static void sortNames (String[] names, int n)
{
    boolean sorted = false;
    while (!sorted)
    {
        sorted = true;
        for (int i=0; i<n-1; i++)
        {
            if (names[i].compareTo(names[i+1])>0)
            {
                sorted = false;
                String temp = names[i];
                names[i] = names[i+1];
                names[i+1] = temp;
            }
        }
    }
}
```

Can you finish the program?