CSC111 Lab

Arrays – Lab II

---- Lab 10 part 2 ---

Create a project **Lab10 (**Or use same project from part 1**).**

Q1) Write a Java program that will declare an array of integers **ar** with **Max_Size** of 50. Then, read array elements from the user and store them in **ar.** You must stop reading integers when the user enters -1.
Then, do the following:

- Print the **size** of the array.
- Print all elements in the array.
- read a number **x** and check if it is in **ar** or not. If it is in the array then print a suitable message stating that it is in the array and its location, otherwise, print a message stating that its not one of the elements in the array.
- Shift all the elements of the array one position to the right.
- Print all elements in the array.

Call your class **arrayOps**.

Sample Run:

```
Please enter the numbers: 5 99 3 10 8 109 -1
The array is of size: 6
The elements of the array are: 5 99 3 10 8 109
Please enter the search number: 23
Sorry. Number 23 is not in the Array.
After shifting, The elements are: 109 5 99 3 10 8
```

Sample Run:

```
Please enter the numbers: 5 99 3 10 8 109 -1
The array is of size: 6
The elements of the array are: 5 99 3 10 8 109
Please enter the search number: 99
The number 99 is at Index 1
After shifting, The elements are: 109 5 99 3 10 8
```

Q2) Improve **arrayOps** from the previous question by converting the print, search and shift operations into methods:

- **static void printArray(int[] a, int size)** Prints all the elements in the array **a**.
- **ststic int search(int[] a, int x, int size)** Searches inside **a** for **x.** If found, the method return its location, otherwise it returns -1.
- **static void shiftRight(int[] a, int size)** Shifts the array **a** to the right one position.

You must also improve the main program. After reading the array, create a suitable menu that gives the user options to print, search or shift the array. Print the elements of the array after each operation.
You must do this repeatedly until the user chooses to stop.

Q3) Write a Java program that will read **n** integers from the user and store them in an array **ar**. Then, read two integers, **x** and **y**, both less than **n** and switch the two array elements located at indecies **x** and **y**. (i.e. switch **ar**[x] and **ar**[y] with each other. This is called swap)
Call your class **arraySwitch**.

Hint:
To swap two array elements at indecies **i** and **j**, you'll need a temporary variable **temp**:

```
    temp = ar[j];
    ar[j] = ar[i];
    ar[i] = temp;
```

Where **temp** is of the same type as **ar** elements.

Sample Run:

```
Please enter how many numbers: 7
Please enter the numbers: 2 4 9 2 -4 -12 3
The elements of the array are: 2 4 9 2 -4 -12 3
Please enter the locations to switch: 2  5
The elements of the array are: 2 4 -12 2 -4 9 3
```

Q4) Write a Java program that will read **n** integers from the user and store them in an array **ar**. But make sure when adding a new element that it is unique and not in the array. If the number already exists, do not add it. Finally, print size and all elements of the array.
Call your class **uniqueArray**.

Sample Run:

```
Please enter how many numbers: 7
Please enter the numbers: 5 99 3 10 3 8 109
The number 3 is a repeat and was not added to the array.
The array is of size: 6
The elements of the array are: 5 99 3 10 8 109
```

Extra:

1) Convert **shiftRight(int[] a, int size)** from **Q2** into **shiftRight(int[] a, int size, int n)** where you shift the array **a** to the right **n** times.

2) Implement **void shiftLeft(int[] a, int size)** Which shifts the array to the left one position.