

Homework4:

Submit your work (the .java source code files ONLY, not the compiled .class files!) through the “Homework4” link on Brightspace. You may submit an unlimited number of times; we will only grade the last/latest submission attempt, but be sure to attach all of your files to each submission attempt. Be sure to include your name and Stony Brook ID number in a comment at the beginning of each file that you submit.

Due: Wednesday, November 6, 11:59 pm **Total: 35 points (25 + 10 points)**

Submission Instructions: Name your java classes for this assignment as:

Problem1: ArrayUtils.java

Problem2: InterleaveArrays.java

1. Create a Java file named `ArrayUtils.java` and follow the steps below:
 - a. (5 points) Define a method, named `randomArray` with two parameters: 1) an integer indicating the size of the array to be created and 2) a positive integer indicating the upper limit for the range of random numbers to be generated. If the second parameter is 101, then it would mean that the random numbers will be in the range of 0 and 100 inclusive. This method should create an array of the given size and fill it with random numbers between the given range. The function returns the array created as its return value. In your main call `randomArray` with two actual arguments: 100 as the size and 21 as the upper limit, and store the returned array into a local variable in main.
Example:
`randomArray(5, 21)` returns `[3, 16, 2, 3, 19]`
 - b. (5 points) Define a method named `print` with one parameter of type array of integers, which outputs the elements of the given array to the standard output device, i.e., the computer screen. In your main call `print` with the local variable that you chose in the step above to see if the elements in the array are printed okay. You will want to use this `print` function whenever you want to see the elements of an array containing integers. While testing your program, include a call to this function wherever useful, but comment out all the calls to this function in the final version that you hand in.
Example:
`print(new int[]{1, 2, 3, 4})` prints `[1 2 3 4]`
 - c. (5 points) Define a method, named `arraySum` with one parameter of type array of integers, which returns the sum of all the elements in the array. In your main print the average as a double of all the numbers in the array that you obtained in the first step above. To compute the average, you must use `arraySum` that you defined earlier. Make sure to add annotations to the output so that it will be meaningful for people seeing the program run.

Example:

```
arraySum(new int[]{1, 2}) returns 3
```

- d. (5 points) Define a method named `contains` with two formal parameters: 1) an array of integers and 2) a single integer. The method returns the index of the array where the first occurrence of the second argument is found, if that number is contained within the array. If the second argument is not contained in the first array argument, the method returns `-1`. Do not use the existing `indexOf` method for solving this problem, you are writing your own solution. In your main call `contains` with the array obtained in the first step above (the random array with 100 as the size and 21 as the upper limit) and 12 as the second argument, and prints the result.

Example:

```
contains(new int[]{1,2,3,4}, 2) returns 1
contains(new int[]{1,2,3,4}, 5) returns -1
```

- e. (5 points) Define a method named `countMultiplesOf` and add it to the `ArrayUtils` class. The method should take two parameters: 1) an array of integers and 2) an integer. This method returns the count of the integers in the array that are multiples of the second parameter. For example, 8 is a multiple of 2, but not a multiple of 3, and zero is a multiple of any number. Hint: I suggest that you also define and use an auxiliary function that tests if a number is a multiple of another number and returns a boolean value. In your main call `countMultiplesOf` with the array obtained in the first step. Use 7 as the second argument and print the result.

Example:

```
countMultiplesOf(new int[]{1,2,3,4}, 2) returns 2 (because 2
and 4 are multiples of 2).
```

2. (10 points) `InterleaveArrays.java`: Write a method `public static int[] interleaveArrays(int[] array1, int[] array2)` that takes in two arrays of integers. Create and return a new array by interleaving the array together. Once an array is empty, use all the elements from the remaining array. Include in your program the test cases listed below and add at least 2 more.

Example:

```
interleaveArrays(new int[]{1,3,5}, new int[]{2,4,6}) returns
[1, 2, 3, 4, 5, 6]
interleaveArrays(new int[]{10,20,30,40,50,60,70,80}, new
int[]{2,4,6,8}) returns [10, 2, 20, 4, 30, 6, 40, 8, 50, 60,
70, 80]
interleaveArrays(new int[]{1,3,5}, new int[]{2,4,6,8,10})
returns [1, 2, 3, 4, 5, 6, 8, 10]
```