CS-141 Basic Array Problems

1-Dimensional Arrays

1. Declare and create an array of 25 integers called `myNums`. Initialize the array by setting the ith element to the value 2*i.
2. Print all the elements of `myNums` using each of the following approaches:
    a. the `Arrays` class
    b. a *for-loop*
    c. an *enhanced for-loop*.
3. Write a method which prints the elements (one element per line) of `myNums` using either 2b or 2c. Test your method by calling it from main().
4. Write a method which returns a String containing all the elements of `myNums` which are divisible by the number 3. Elements should be separated by spaces. Test your method by calling it from main().
5. Swap the 5th element of `myNums` with the 10th element.
6. Write a method which swaps specified elements, e.g. its header might look like:
   ```
   public void swap(int[] myArray, int i, int j)
   ```
   Why won't it work to write the method as follows:
   ```
   public void swap(int myArray[i], int myArray[j])
   ```
   What is the difference between pass by value and pass by reference?
7. Declare and create an array of 20 numbers (doubles) called `nums.` Initialize each element to a random number. Compute their average.
8. Insert an new value into the 5th index of `nums`, shifting items to make room. Note, the last item will be lost.
9. Remove the 10th index in the array `nums`, shifting items to remove the hole. Place the removed item into the last position of the array.
10. Use the Arrays class to sort `nums`. Print the array before and after the sort.
11. Copy `nums` into an new array using the following approaches:
    a. The Arrays.copyOf() method
    b. Create a second array of the same size and use a loop to set the values of the second array to be the same as `nums`.
12. Write a method which creates and returns an array of 20 random integers.
13. Write a method which takes an integer array as a parameter and doubles the value of each of its elements.

2-Dimensional Arrays

1. Declare and create (using curly brackets {..}) a 2D array of Strings called `names` with 2 rows and 3 columns. Use whatever values you want.
2. Declare and create a 2D array of `doubles` called `cells` with 4 rows and 8 columns. Using loops,

set the value of each element of `cells` to be equal to the product of its row and column.

3. Try to answer the following questions about `cells`: What type of variable is `cells`? What type is `cells[2]`? What type is `cells[1][2]`? What is the value of `cells.length`? What is the value of `cells[2].length`? Does `cells[2][3].length` make any sense? Why or why not?
4. Compute and print the average of the values in each row of `cells`.
5. Compute and print the average of the values in each column of `cells`.
6. Compute and print the average over *all* the values in `cells`.
7. Suppose you are writing a solitaire card game which begins with 7 piles of cards aligned in a row. The ith pile contains i+1 cards. Create a 2D array to represent these cards. Set the value of each card randomly to a value in the range 0 to 51.

ArrayLists

1. Declare and create an `ArrayList` of `Strings` called `animals`.
   a. Add animal names to the list (e.g. ant, aardvark, cat, crow, snake, dog, zebra, cheetah, coyote, duck, dingo, deer).
   b. Use a loop to print out the resulting elements in the `animals`.
   c. Use a loop to remove all animals whose names begin with `'a'` or `'c'`.