HW0: Getting to Know You and Java Practice

Read Me Before Starting

This is the very first homework in CSC 4520. The goal is to learn more about you as an individual and explore your knowledge in programming and Java. **This assignment is due on September, 6th, 2022 at 11:59PM Eastern Time.**

Submitting This Assignment

On iCollege

Grading and Corrections

We've provided some example inputs and outputs to help you test your code and check your work. Producing the correct output for these examples is necessary but not sufficient for receiving full credit. There may be other test cases that we use in grading--you should consider writing your own examples.

Showing Your Work

There are three ways of showing your work. Any of these ways will help you earn "effort" credit. Following these practices will also help us verify that your work is your own. This helps you avoid being falsely accused of academic dishonesty.

1. You should add comments that explain the key ideas behind your approach. We've added placeholders in HW0.java where you can do this.
2. You may also add additional test cases in order to ensure that your code is 100% correct.

Remember that you can consult outside resources and work with other students as long as you write up your own solutions and cite any links or people you received help from within citations.txt. See syllabus/collaboration for details.

Q1: maxOfArray (3 points)

Write maxOfArray, which takes in an array of integers and returns the largest integer within the array. If the array is empty, throw an IllegalArgumentException. Some examples:

```
int testResult1 = maxOfArray(new int[] {1, 3, 4, 5, 2});
int testResult2 = maxOfArray(new int[] {-1, -3, -4, -5, -2});

System.out.println(testResult1); // should output 5
System.out.println(testResult2); // should output -1

maxOfArray(new int[] {}); // Should throw IllegalArgumentException
```

Q2: twoSum (3 points)

Write twoSum, which takes in an array of integers and a target sum, and returns a 2-element array that represents two *distinct* indices of elements that sum up to the target value. Some examples:

```
int[] testResult3 = twoSum(new int[] {0, 2, 3, 4, 5}, 6);
int[] testResult4 = twoSum(new int[] {1, 2, 3, 4, 5}, 10);

System.out.println(Arrays.toString(testResult3)); // should output [1, 3]
System.out.println(Arrays.toString(testResult4)); // should output [-1, -1]
```

- In the first example, `arr[1] + arr[3] = 2 + 4 = 6`.
- In the second example, we returned `[-1, -1]` because there are not two distinct elements within the array that sum to 10 (you can't use 5 twice).

Q3 (4 points)

Write add, which given two numbers represented as Lists of single-digit integers, returns their sum as a list of integers. Some examples:

```
List<Integer> testResult5 = add(Arrays.asList(1, 2, 3), Arrays.asList(2, 4, 2));

List<Integer> testResult6 = add(Arrays.asList(9, 9, 9), Arrays.asList(1));

// 123 + 242 = 365
// [1, 2, 3], [2, 4, 2] => [3, 6, 5]
System.out.println(testResult5); // should output [3, 6, 5]

// 999 + 1 = 1000
// [9, 9, 9], [1] => [1, 0, 0, 0]
System.out.println(testResult6); // should output [1, 0, 0, 0]
```