COMPUTER SCIENCE A
SECTION II

Time—1 hour and 45 minutes

Number of questions—4

Percent of total score—50

Directions: SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.

Notes:

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

1. This question involves reasoning about one-dimensional and two-dimensional arrays of integers. You will write three static methods, all of which are in a single enclosing class, named `DiverseArray` (not shown). The first method returns the sum of the values of a one-dimensional array; the second method returns an array that represents the sums of the rows of a two-dimensional array; and the third method analyzes row sums.

    (a) Write a `static` method `arraySum` that calculates and returns the sum of the entries in a specified one-dimensional array. The following example shows an array `arr1` and the value returned by a call to `arraySum`.

		`arr1`			Value returned by `arraySum(arr1)`
0	1	2	3	4	
1	3	2	7	3	16

WRITE YOUR SOLUTION ON THE NEXT PAGE.

GO ON TO THE NEXT PAGE.

Complete method arraySum below.

```
/** Returns the sum of the entries in the one-dimensional array arr.
 */
public static int arraySum(int[] arr)
```

Part (b) begins on page 4.

GO ON TO THE NEXT PAGE.

Question 1: Diverse Array

Part (a) `arraySum` 2 points

Intent: *Compute and return sum of elements in 1D array* `arr`*, passed as parameter*

+1 Accesses all elements of `arr`, (*no bounds errors on* `arr`)

+1 Initializes, computes, and returns sum of elements

Part (b) `rowSums` 4 points

Intent: *Compute and return 1D array containing sums of each row in the 2D array* `arr2D`*, passed as parameter*

+1 Constructs correctly-sized 1D array of ints

+1 Accesses all words in `arr2D` (*no bounds errors on* `arr2D`)

+1 Computes sum of row in `arr2D` using `arraySum` and assigns to element in 1D array

+1 Returns 1D array where kth element is computed sum of corresponding row in 2D array for all rows

Part (c) `isDiverse` 3 points

Intent: *Determine whether* `arr2D`*, passed as parameter, is diverse*

+1 Computes and uses array of row sums from `arr2D` using `rowSums`

+1 Compare all and only pairs of row sums for equality (*No bounds errors on row sums array; point not awarded if no adjustment when compares any row sum with itself*)

+1 Returns `true` if all compared row sums are different and `false` otherwise (*point not awarded for immediate return*)

Question-Specific Penalties

-1 (g) Uses `getLength/getSize` for array size

-1 (y) Destruction of persistent data (`arr` *or* `arr2D`)

AP® COMPUTER SCIENCE A
2015 CANONICAL SOLUTIONS

Question 1: Diverse Array

Part (a):

```
public static int arraySum(int[] arr){
       int sum=0;
       for (int elem : arr){
            sum += elem;
       }
       return sum;
}
```

Part (b):

```
public static int[] rowSums(int[][] arr2D){
        int [] sums=new int[arr2D.length];
        int rowNum=0;
        for(int[] row : arr2D){
            sums[rowNum]=arraySum(row);
            rowNum++;
        }
        return sums;
    }
```

Part (c):

```
public static boolean isDiverse(int[][] arr2D){
        int [] sums=rowSums(arr2D);
        for (int i=0; i < sums.length; i++){
            for (int j=i+1; j < sums.length; j++){
                if (sums[i]==sums[j]){
                    return false;
                }
            }
        }
        return true;
    }
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.