AP CS A pset **Name**_____
Lesson 5.6

The `divBySum` method is intended to return the sum of all the elements in the `int` array parameter `arr` that are divisible by the `int` parameter `num`. Consider the following examples, in which the array `arr` contains `{4, 1, 3, 6, 2, 9}`.

- The call `divBySum(arr, 3)` will return 18, which is the sum of 3, 6, and 9, since those are the only integers in `arr` that are divisible by 3.
- The call `divBySum(arr, 5)` will return `0`, since none of the integers in `arr` are divisible by 5.

Complete the `divBySum` method using an enhanced `for` loop. Assume that `arr` is properly declared and initialized. The method must use an enhanced `for` loop to earn full credit.

```
/** Returns the sum of all integers in arr that are divisible by
num
* Precondition: num > 0
*/
public static int divBySum(int[] arr, int num)
```

2. An array of `String` objects, `words`, has been properly declared and initialized. Each element of `words` contains a `String` consisting of at least 3 lowercase letters (a–z).

Write a code segment that uses an enhanced `for` loop to print all elements of `words` that end with `"ing"`. As an example, if `words` contains `{"ten", "fading", "post", "card", "thunder", "hinge", "trailing", "batting"}`, then the following output should be produced by the code segment.

```
fading
trailing
batting
```

Write the code segment as described above. The code segment must use an enhanced `for` loop to earn full credit.

AP CS A pset **Name**_____
Lesson 5.6

3. The `Vocab` class (**see handout**) contains methods used to analyze words in terms of their presence in a controlled vocabulary. You will write two methods of the `Vocab` class.

The `countNotInVocab` method returns an `int` that contains the number of words in its parameter `wordArray` that are not found in the instance variable `theVocab`.

A helper method, `findWord`, has been provided. The `findWord` method searches for an individual string in `theVocab`, returning `true` if an exact match between its `String` parameter and an element of `theVocab` is found, and returning `false` otherwise.

(a) Write the `countNotInVocab` method. Assume that there are no duplicates in `wordArray`. You must use `findWord` appropriately to receive full credit.

```
/** Counts how many strings in wordArray are not found in
theVocab, as described in
* part (a).
*/
public int countNotInVocab(String[] wordArray)
```

(b) Write the `notInVocab` method (**see handout**). Assume that there are no duplicates in `wordArray`. You must call `findWord` and `countNotInVocab` appropriately in order to receive full credit.

```
/** Returns an array containing strings from wordArray
not found in theVocab,
* as described in part (b).
*/
public String[] notInVocab(String[] wordArray)
```

AP CS A pset **Name**_____
Lesson 5.6