Problem #1

- Write a program that has the class name `Problem1` and that has the `main` method.

- Write a method named `multipleOfIndices` that takes one parameter, a positive integer array `arr` and returns a boolean array.

- For every integer in the integer array, the program should check if the integer is a multiple of the index it is in, and assign the boolean value as an element for boolean array at that index. Note that if the index is 0th and 1st, then return `true` if the remainder is equal to the index when divided by 10 else return `false`.

- As a reminder, a number `m` is a multiple of `n` if `m` can be evenly divided into `n`. For example, 24 can be divided into 3 evenly, therefore 24 is a multiple of 3, so the element would get a value of `true` if 24 is in a 3rd index.

- Create a `printArray` method that takes a boolean array as a parameter and prints out the elements of the array on the same line separated by a space.

- Several sample runs are provided for you below. Your output must be formatted **exactly** like the sample runs below. Use the sample usages in the `main` method to test your code

Sample Method Usage	Return Value
`int[] a1 = { 1, 21, 5, 9, 12, -50, 47 };` `boolean[] b1 = multipleOfIndices(a1);`	`{ false, true, false, true, true, true, false }`
`int[] a2 = {5, 3, 77, 34, 43 };` `boolean[] b2 = multipleOfIndices(a2);`	`{ false, false, false, false, false }`
`int[] a3 = { 30, 22, 42, 8, 15, 27, 6 };` `boolean[] b3 = multipleOfIndices(a3);`	`{ true, false, true, false, false, false, true }`
`int[] a4 = { 10, 51, 34, 69, 44, 95 };` `boolean[] b4 = multipleOfIndices(a4);`	`{ true, true, true, true, true, true }`

Problem #2

- Write a program that has the class name `Problem2` and that has the `main` method. Leave the `main` method empty for now.

- Write a method named `greaterThanSum` that takes one parameter, an integer array `a` and returns a new integer array `x`.

- The method finds all the terms of array that are greater than the sum of all previous terms of the sequence. If there is no such elements, you can return an empty array.

- Several sample usages are provided for you below. Use the sample usages in the `main` method to test your code. Create a `printArray` method that takes an integer array as a parameter and prints out the elements of the array on the same line separated by a comma and space.

Sample Method Usage	Return Value
`int[] a1 = { 1, 4, 16, -19, -12, 2, 5 };` `int[] x1 = greaterThanSum(a1);`	`{1, 4, 16, 2, 5 }`
`int[] a2 = { -1, -2, -4, -12 };` `int[] x2 = greaterThanSum(a2);`	`{ }`
`int[] a3 = { 29, -10, 22, 5, -15, 19, 62};` `int[] x3 = greaterThanSum(a3);`	`{29, 22, 62 }`
`int[] a4 = { 5, 8, 17, 50};` `int[] x4 = greaterThanSum(a4);`	`{5, 8, 17, 50 }`