BinarySearcher Functional Requirements:

The BinarySearcher application allows you to create an instance by taking in an integer array along with an integer key to search for its existence inside the array.

It provides methods to perform a search with the searchKey() method. The application is expected to behave functionally as per the below rules.

1. The input array provided will be automatically sorted when stored.
2. The input stored in the application can always be accessed through the `getInputItems` method. And can be set using the `setInputItems()` methods.
3. The search key in the application can always be accessed through the `getSearchKey` method. And can be set using the `setSearchKey()` methods
4. The search could be performed with the `doSearch()` method
5. The results from the `doSearch()` method obeys the following rules
    a. If the Key Exists in the array:
        - It always gives a positive result as the position at which the element occurs in the list (Index is 1 based)
        - **Example** when you search for **3** in the list **[10, 1 , 3, 5]** - > returns **2**, as the position of **3** in the sorted input array **[1 , 3, 5, 10]** will be 2.
    b. If the Key does not exist in the input list:
        i. If the key is not found in the input list the result is always negative and it is the negative value of its insertion point in the input list.
        ii. **Example** when i search for 9 in the list **[10, 1 , 3, 5] - > returns (- 4)** as the insertion position of 9 would be after 5 in the sorted array **[1 , 3, 5, 10]** which is **4** and since the element was not found its negated value is returned.

Tasks to be Performed:

You are the automation tester assigned to the test case design and automating the scenarios to ensure a complete coverage of the application behavior. You are supposed to write your test case specification using the BDD style gherkin syntax implemented using the cucumber-java framework. Create a feature file accordingly reflecting the feature tested and write gherkin style specification to be able to check the below scenarios.

1. When the application is launched with various input lists and keys to search, the application **always** stores the input list in sorted order.
2. Ensure that with various inputs to the application , you always get back the exact key you had specified when accessing it from the app.
3. Ensure that even after launching the application the input items could be modified with a different list and the application will store it in sorted order. (Hint: Use scenario outline with the list parameterised)
4. With the correct input cases check that the implementation is correct for existing keys.
5. With the correct input cases check that the implementation is correct for non existing keys.
6. Have implementations mechanisms to your scenarios that you will have the capability to run only the non existing scenarios / rules or the item found scenarios (Hint: tags)

RandomIntegerGenerator Functional Requirements:

The RandomIntegerGenerator app is used to generate an array of pseudorandom integers. The class uses a seed if provided or uses a random seed if not specified.

1. If two instances of RandomIntegerGenerator are created with the same seed, and the same sequence of method calls is made for each, they will generate and return identical sequences of numbers.
2. `generateNumbers(int size)` Returns an Integer Array of the given size pseudo random int values
3. `generateNumbers(int size, int randomNumberOrigin, int randomNumberBound)` Returns an array of random int values, each conforming to the given origin (inclusive) and bound (exclusive).
4. `getSingleValue()` returns an integer from the random pool of integers.

Tasks to be Performed:

You are the automation tester assigned to the test case design and automating the scenarios to ensure a complete coverage of the RandomIntegerGenerator application behavior. You are supposed to write your test case specification using the BDD style gherkin syntax implemented using the cucumber-java framework. Create a feature file accordingly reflecting the feature tested and write gherkin style specification to be able to check the below scenarios.

1. The application could be created without specifying a seed value.
2. The application could be created by specifying a seed value.
3. When you create 2 instances of the application with the same seed value, the returned values from both the instances are identical. (use a scenario template)
4. Write scenarios to check that the app is capable of producing various lengths of data values when you specify the size. (use data table)
5. Write scenarios to check that when you specify the size , origin and bound the resulting values are obeying the rules as in functional Requirement 3.
6. Have implementations mechanisms to your scenarios that you will have the capability to run only the single value generation and multiple value generation scenarios (Hint: tags)

SimpleTextScanner Functional Requirements:

SimpleTextScanner is an application that can be created by specifying an input string along with a **delimiter** (which by default matches whitespace) and a **regular expressions pattern**. Once created the app can parse primitive types and strings using the provided regular expressions pattern.

SimpleTextScanner breaks its input into tokens using the specified delimiter pattern, The resulting tokens may then be converted into values of different types using the various `next` methods.

TODO: <>