

1h 49m  
left

ALL



1

2

3

4

## 2. Java Threads: Strings Collection

Data integrity when a shared resource is accessible to more than one thread is important. In this challenge, create a thread-safe shared list of strings. Create the *StringsCollection* class that has a data member *List<String> stringsCollection*. The class should implement the following two methods:

1. *void addString(String string)* adds the given string to the strings collection.
2. *List<String> getStringsCollection()* returns the strings collection.

The locked stub code in the editor validates the correctness of the *StringsCollection* class implementation by adding strings to the collection using threads. The locked stub code prints the total number of strings and the number of non-null strings in the collection, each on a separate line.

### Constraints

- $1 \leq threadsCount \leq 10$
- Each thread adds no more than  $10^5$  strings.

### ► Input Format For Custom Testing

### ▼ Sample Case 0

#### Sample Input 0

| STDIN | Function                    |
|-------|-----------------------------|
| ----- | -----                       |
| 3     | → threadsCount = 3          |
| 2     | → thread 0 stringsCount = 2 |
| 3     | → thread 1 stringsCount = 3 |
| 2     | → thread 2 stringsCount = 2 |

#### Sample Output 0

7  
7

#### Explanation 0

There are three threads:

- The first thread adds two strings "11" and "12".
- The second thread adds three strings "21", "22", and "23".