

#1- Audit logins to web service: Question description (Java 8)

From the call: This question focuses on multi-threading and execution service. Candidates can write their own separate thread or they can have Java provide the thread through an execution service.

ABC Corp has a popular rest API that is used by millions of customers.

ABC Corp has a change requirement to audit every login to this API.

To facilitate this, a method `writeAuditLog` has been implemented. This method takes two parameters, a log message, and a user id. The log message should be of the format "Login success at <iso_date_timestamp>".

This audit feature has to be implemented without impacting the response time of the existing api. The writes to the audit database take between 1 and 500 milliseconds.

Your task is to implement the `auditLogin` method with minimal impact to the existing response time of the API.

#2- Movie Titles: Question description (Rest API, GET, Back-end Development, JSON, Role based skills)

From the call: API / JSON- This questions consists of one JSON library. Test takers can use the JDK API or online tools to complete the test. If they don't remember Networking API's they can look at the API documentation, a lot of this is now handled by libraries like Spring.

Query a JSON response and print certain results.

To solve this challenge, write an *HTTP GET* method to retrieve information from a particular movie database. Complete the function in the editor; it has one parameter: a string, *substr*. The function must perform the following tasks:

1. Query <https://jsonmock.hackerrank.com/api/movies/search/?Title=substr> (where *substr* is the value of *substr*). The query response from the website is a JSON response with the following five fields:
 - `page`: The current page.
 - `per_page`: The maximum number of results per page.
 - `total`: The total number of such movies having the substring *substr* in their title.
 - `total_pages`: The total number of pages which must be queried to get all the results.
 - `data`: An array of JSON objects containing movie information where the Title field

denotes the title of the movie. Note that this field is paginated so, in order to incorporate pagination, you must query <https://jsonmock.hackerrank.com/api/movies/search/?Title=substr&page=pageNumber>, where *pageNumber* is an integer denoting the page you would like to view (e.g., 1, 2, etc.).

2. Create an array of strings named *titles* to store total elements. For each page of results, store the Title of each movie in the *titles* array.
3. Sort *titles* in ascending order and return it as your answer.

Input Format

A single string, *substr*, denoting the substring you must query for.

Output Format

Return an array of strings corresponding to movie titles with *substr* in their Title, sorted in ascending order.

#3- Maximum difference in Array: (data structures, algorithms, arrays, core skills, problem solving)

From the call: More straight forward Java coding question. Our candidates have said this is one of the easier questions.

The *maximum difference* between elements in some array, *a*, is defined as the largest difference between any $a[i]$ and $a[j]$ where $i < j$ and $a[i] < a[j]$. For example, if $a = [4, 1, 2, 3]$, the maximum difference would be $a[3] - a[1] = 3 - 1 = 2$ because this is the largest difference between any two elements satisfying the aforementioned criteria.

Complete the *maxDifference* function in the editor below. It has 1 parameter: an array of integers, *a*. It must return an integer denoting the maximum difference between any pair of elements in *a*; if no such number exists (e.g., if *a* is in descending order and all $a[j] < a[i]$), return -1 instead.

Input Format

Locked stub code in the editor reads the following input from stdin and passes it to the function:

The first line contains a single integer, *n*, denoting the number of elements in array *a*.

Each line *i* of the *n* subsequent lines (where $0 \leq i < n$) contains a single integer describing element $a[i]$.

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $-10^6 \leq a[i] \leq 10^6 \forall i \in [0, n - 1]$

Output Format

The function must return an integer denoting the maximum difference in *a*. This is printed to stdout by locked stub code in the editor.

#4- Animal Inheritance (general programming, inheritance, polymorphism, abstraction, java, C++, C#, OOP, Java 8, core skills)

The locked code in the editor does the following:

1. Declares an abstract class named `Animal` with the implementations for `getIsMammal()` and `getIsCarnivorous()` methods, as well as an abstract method named `getGreeting()` .

2. Creates `Dog`, `Cow` , and `Duck` objects.

3. Calls the `getIsMammal()` , `getIsCarnivorous()` , and `getGreeting()` methods on each of these respective objects.

A UML diagram of `Animal`, `Dog`, `Cow`, and `Duck` classes.

Recall that `-` denotes private, `+` denotes public, and `#` denotes protected .

Complete the code in the editor below by writing the following:

1. Three classes named `Dog` , `Cow` , and `Duck` that inherit from the `Animal` class.

2. No-argument constructors for each class that initialize the instance variables inherited from the superclass.

3. Each class must implement the `getGreeting()` method:

For a `Dog` object, this must return the string `ruff` .

For a `Cow` object, this must return the string `moo` .

For a `Duck` object, this must return the string `quack` .

#5- In Memory Contacts Data Store (Java 8, coding)

ABC Corp has a requirement to provide an in-memory contacts

data store. This requirement has two features:

1. The store should implement the provided interface with a method "command", a method "find", and a method "all".

The "command" method is used to ADD/UPDATE/DELETE contacts.

The "find" method is used to find a contact given a partial name match. The find should maintain the case when performing the partial match. Use "contains" to match the name string.

The "all" method returns a list of all the contacts currently in the in memory store.

2. Implement a Comparator for the Contact data class which sorts on the nationalId. NationalId will be unique.