**Hacker Earth – Java Certification**

**Microservices, Springboot Questions**

1. Microservices, Spring and Springboot

1. How to do SSO implementation using Spring Boot? Which pattern we can use from Microservices?
2. What is a CDC? What is a Client Certificate?
3. Explain Generate API documentation for a Spring Boot RESTful API by using swagger?
4. What is the difference between idempotent and safe HTTP methods?
5. How do you implement observability in microservices architecture?
6. What do you understand by auto-configuration? How can you write a custom auto-configuration?
7. What is Command Query Responsibility Segregation (CQRS) design pattern? Cross-Functional Requirements to consider during the design of a Microservice?
8. Define HttpMessageConverter in terms of Spring REST ?

1. How do you limit a Spring controller method to handle only HTTP PUT operation requests? Can you write method coding?

1. What is Restlet? Write a code example of a Microservice in Restlet?

1. @WebMvcTest(value = ToTestController.class, secure = false) – What is use of following code?

1. How do you handle backward compatibility in microservices? How do you handle data consistency across microservices?
2. Is it possible to make asynchronous requests in JAX-RS?

1. How do you implement observability in microservices architecture?

1. How to use crud repository in spring boot? Explain with example?

1. What is the difference between Mean Time to Repair (MTTR) and Mean Time between failures (MTBF)?

1. How to do SSO implementation using Spring Boot? Which pattern we can use from Microservices?

1. Why some big companies use API keys for providing access to public APIs? What is Confused Deputy Problem in security context?
2. Solid principles explain polymorphism in your application how to convert heterogenous list to array cyclic dependency in spring how do u navigate service layer exception to controller what are annotations worked in springboot?
3. How do you ensure data privacy and protection in microservices? What are the key factors to consider when migrating to microservices?
4. What is CORS in Spring Boot? How to enable CORS in Spring Boot? What is GZIP? How to implement it using Spring Boot? How to enable HTTP response compression in Spring Boot?
5. How will you implement service monitoring and service logging in a microservices architecture?
6. What are different types of Tests for Microservices? What is a message broker service in java?
7. How to run spring-boot jar from the command line? Write the commands
8. Write a program for following, Search all the hotels closest to the city center. /search/{cityId}. The city object has latitude and longitude and the hotel object also has latitude and longitude. The distance between two can be calculated using the haversine formula, (The question says we can use the internet to calculate the haversine formula) by Spring boot.
9. What is the code used to integrate Thymeleaf in Spring Boot?  how to use thymeleaf?
10. Mention the advantages of the YAML file than Properties file and the different ways to load YAML file in Spring boot.
11. What function would you use to enable HTTP/2 support in Spring? Can you explain what happens in the background when a Spring Boot Application is “Run as Java Application”?
12. How does path=”sample”, collectionResourceRel=”sample” work with Spring Data Rest?
13. How do you create a Spring Boot project using boot CLI? Write the CLI Commands?
14. What annotations are used to create an Interceptor? Write an Example methods?
15. You have a complex business logic that involves multiple steps and dependencies, and you need to implement transaction management to ensure data integrity. How would you configure declarative transaction management using Spring’s transaction management support?
16. You are building a message-driven application using Spring Boot and need to implement asynchronous communication using message brokers such as RabbitMQ or Apache Kafka. How would you configure and use Spring’s support for messaging to send and receive messages asynchronously?
17. Please explain the Bean lifecycle in Spring Bean Factory Container? Write bean configuration?
18. You are building a RESTful API using Spring Boot and need to implement API documentation and testing using OpenAPI (formerly Swagger) and JUnit. How would you configure and use Springfox or Spring REST Docs to generate API documentation and write API tests?
19. Write Spring JWT token for the Username and Password?
20. A company wants to use Spring Boot in a web application which should use JPA as a data base abstraction. The unit tests should be run on H2 data base while the production run on a MySQL Database. Can you explain this.
21. Explain Generate API documentation for a Spring Boot RESTful API by using swagger?
22. What do you understand by auto-configuration? How can you write a custom auto-configuration?
23. How to run spring-boot jar from the command line? Write the commands.

**Hacker Earth – Java Certification**

1. SQL Query

1. Given the CITY and COUNTRY tables, query the sum of the populations of all cities where the CONTINENT is 'Asia'.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The CITY and COUNTRY tables are described as follows:

City

|  |  |
| --- | --- |
| **Field** | **Type** |
| ID | Number |
| Name | Varchar2(17) |
| Country Code | Varchar2(3) |
| District | Varchar2(20) |
| Population | Number |

A screenshot of a computer

Description automatically generated with low confidence

1. There are two data tables with employee information: EMPLOYEE and EMPLOYEE\_UIN. Query the tables to generate a list of all employees who have been employed fewer than three years in order of NAME, then of ID, both ascending. The result should include the UIN followed by the NAME. While the secondary sort is by ID, the result includes UIN but not ID.

Guideline : Join the tables to get UIN. Filter results to TIME < 3 and sort ascending by name, id.

Schema

EMPLOYEE

Name Type Description

ID Integer The ID of the employee. This is a primary key.

NAME String The name of the employee having [1, 20] characters.

TIME Integer The tenure of the employee.

ADDRESS String The address of the employee having [1, 25] characters.

SALARY Integer The salary of the employee.

EMPLOYEE\_UIN

Name Type Description

ID Integer The ID of the employee. This is a primary key.

UIN String The unique identification number of the employee.

1. Query the names of all American cities in CITY with populations larger than 120000. The CountryCode for America is USA.

1. Query a list of CITY names from STATION with even ID numbers only. You may print the results in any order, but must exclude duplicates from your answer.

1. Query the list of CITY names starting with vowels (i.e., a, e, i, o, or u) from STATION. Your result cannot contain duplicates

1. Query the list of CITY names from STATION which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

1. Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is ‘Africa’. Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

1. Write a query that prints a list of employee names (i.e.: the name attribute) from the Employee table in alphabetical order.

Input Format

The Employee table containing employee data for a company is described as follows:

A picture containing text, screenshot, font, number

Description automatically generated

where employee\_id is an employee's ID number, name is their name, months is the total number of months they've been working for the company, and salary is their monthly salary.

A picture containing text, screenshot, number, font

Description automatically generated

1. Write a query that prints a list of employee names (i.e.: the name attribute) for employees in Employee having a salary greater than per month who have been employees for less than months. Sort your result by ascending employee\_id.

Input Format

The Employee table containing employee data for a company is described as follows:

A picture containing text, font, screenshot, number

Description automatically generated

where employee\_id is an employee’s ID number, the name is their name, months is the total number of months they’ve been working for the company, and salary is their monthly salary.

1. Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The CITY and COUNTRY tables are described as follows:

A screenshot of a computer

Description automatically generated with low confidenceA picture containing text, font, screenshot, number

Description automatically generated

1. Query the list of CITY names from STATION which have vowels (i.e., a, e, i, o, and u) as both their first and last characters. Your result cannot contain duplicates.

1. Given the CITY and COUNTRY tables, query the names of all cities where the CONTINENT is ‘Africa’. Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

1. Write a query that prints a list of employee names (i.e.: the name attribute) from the Employee table in alphabetical order.

Input Format

The Employee table containing employee data for a company is described as follows:

A picture containing text, screenshot, font, number

Description automatically generated

where employee\_id is an employee's ID number, name is their name, months is the total number of months they've been working for the company, and salary is their monthly salary.

A picture containing text, screenshot, number, font

Description automatically generated

1. Write a query that prints a list of employee names (i.e.: the name attribute) for employees in Employee having a salary greater than per month who have been employees for less than months. Sort your result by ascending employee\_id.

Input Format

The Employee table containing employee data for a company is described as follows:

A picture containing text, font, screenshot, number

Description automatically generated

where employee\_id is an employee’s ID number, the name is their name, months is the total number of months they’ve been working for the company, and salary is their monthly salary.

1. Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

Note: CITY.CountryCode and COUNTRY.Code are matching key columns.

Input Format

The CITY and COUNTRY tables are described as follows:

A screenshot of a computer

Description automatically generated with low confidenceA picture containing text, font, screenshot, number

Description automatically generated

**Hacker Earth – Java Certification**

**Core Java Questions**

1. Given a time in -hour AM/PM format, convert it to military (24-hour) time.

Note: - 12:00:00AM on a 12-hour clock is 00:00:00 on a 24-hour clock.

- 12:00:00PM on a 12-hour clock is 12:00:00 on a 24-hour clock.

Example

• Return '12:01:00'.

• Return '00:01:00'.

**Function Description**

Complete the timeConversion function in the editor below. It should return a new string representing the input time in 24 hour format.

timeConversion has the following parameter(s):

• string s: a time in  hour format

Returns

• string: the time in  hour format

Input Format

A single string  that represents a time in -hour clock format (i.e.:  or ).

Constraints

• All input times are valid

**Sample Input**

07:05:45PM

**Sample Output**

19:05:45

1. In this challenge, we test your knowledge of using *if-else* conditional statements to automate decision-making processes. An if-else statement has the following logical flow:

A black screen with a white circle

Description automatically generated with low confidence

**Task**  

Given an integer, , perform the following conditional actions:

* If n is odd, print Weird
* If n is even and in the inclusive range of 2 to 5, print Not Weird
* If n is even and in the inclusive range of 6 to 20, print Weird
* If n is even and greater than 20, print Not Weird

Complete the stub code provided in your editor to print whether or not  is weird.

**Input Format**

A single line containing a positive integer, .

**Output Format**

Print Weird if the number is weird; otherwise, print Not Weird.

1. Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.   You can return the answer in any order.

Example 1

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

1. Given a string s, find the length of the longest substring without repeating characters

Example 1:

Input: s = "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

1. Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order.

A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

A close-up of a phone keypad

Description automatically generated with medium confidence

Example 1:

Input: digits = "23"

Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

1. **Problem Statement -:**Anirudh is attending an astronomy lecture. His professor who is very strict asks students to write a program to print the trapezium pattern using stars and dots as shown below . Since Anirudh is not good in astronomy can you help him?

**Sample Input:**

N = 3

**Output:**

\*\*.\*\*

\*…\*

…..

\*…\*

\*\*.\*\*

1. **Problem Statement –**The math assignment says you will be given numbers, mostly with imaginary additions, that means complex numbers, and you need to add them and tell the answer in your answer script. You told your friend John that you don’t know the addition of complex numbers, so John will write a program, which you can write in order to get the results of addition.

John knows Object oriented programming enough to complete the task.

**Input Format:**   
Three integers a b and c   
Output format:   
First print the complex number a+bi   
Next line print a + bi + c as i2.   
Next line i2+a+bi

**Sample Input:**   
4 5 2

**Sample Output:**   
4 + 5i   
6 + 5i   
10 + 10i

1. You are given the heads of two sorted linked lists list1 and list2.

Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists.

Return the head of the merged linked list.

A picture containing circle, weightlifting

Description automatically generated

Example 1:

Input: list1 = [1,2,4], list2 = [1,3,4]

Output: [1,1,2,3,4,4]

1. Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.

If target is not found in the array, return [-1, -1].

You must write an algorithm with O(log n) runtime complexity.

Example 1:

Input: nums = [5,7,7,8,8,10], target = 8

Output: [3,4]

1. One popular way to read input from stdin is by using the [Scanner class](https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html) and specifying the *Input Stream* as *System.in*. For example:

Scanner scanner = new Scanner(System.in);

String myString = scanner.next();

int myInt = scanner.nextInt();

scanner.close();

System.out.println("myString is: " + myString);

System.out.println("myInt is: " + myInt);

The code above creates a *Scanner* object named  and uses it to read a *String* and an *int*. It then *closes* the *Scanner* object because there is no more input to read, and prints to stdout using *System.out.println(String)*. So, if our input is:

Hi 5

Our code will print:

myString is: Hi

myInt is: 5

Alternatively, you can use the [BufferedReader class](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html%22%20/t%20%22_blank%22%20/t%20%22_blank" \t "_blank).

**Task**  

In this challenge, you must read  integers from stdin and then print them to stdout. Each integer must be printed on a new line. To make the problem a little easier, a portion of the code is provided for you in the editor below.

**Input Format**

There are  lines of input, and each line contains a single integer.

**Sample Input**

42

100



**Sample Output**

42

100

125

1. Java's *System.out.printf* function can be used to print formatted output. The purpose of this exercise is to test your understanding of formatting output using *printf*.

To get you started, a portion of the solution is provided for you in the editor; you must format

and print the input to complete the solution.

**Input Format**

Every line of input will contain a *String* followed by an *integer*.    
Each *String* will have a maximum of  alphabetic characters, and each *integer* will be in the inclusive range from  to .

**Output Format**

In each line of output there should be two columns:    
The first column contains the *String* and is left justified using exactly  characters.    
The second column contains the *integer*, expressed in exactly  digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.

**Sample Input**

java 100

cpp 65

python 50

**Sample Output**

================================

java           100

cpp            065

python         050

================================

**Explanation**

Each *String* is left-justified with trailing whitespace through the first  characters. The leading digit of the *integer* is the  character, and each *integer* that was less than  digits now has leading zeroes.

1. A palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward.



Given a string, print Yes if it is a palindrome, print No otherwise.

**Constraints**

* will consist at most  lower case english letters.

**Sample Input**

madam

**Sample Output**

Yes

1. Write a recursive method that reverses a string. For example, your method should print out "rac" given the word "car".

1. Given an integer, convert it to a roman numeral.

Given the head of a linked list, remove the nth node from the end of the list and return its head.

A picture containing circle, diagram, screenshot, line

Description automatically generated

Example 1:

Input: head = [1,2,3,4,5], n = 2

Output: [1,2,3,5]

Example 2:

Input: head = [1], n = 1

Output: []

1. Define the relationships among association, aggregation, composition, and inheritance.
2. **Problem Statement –**The principal has a problem with repetitions. Everytime someone sends the same email twice he becomes angry and starts yelling. His personal assistant filters the mails so that all the unique mails are sent only once, and if there is someone sending the same mail again and again, he deletes them. Write a program which will see the list of roll numbers of the student and find how many emails are to be deleted.

**Sample Input:**   
6   
1   
3   
3   
4   
3   
3

**Sample Output:**   
3

1. **Problem** A famous Italian restaurant allows guests to enter only if they are present in pairs and the sum of the wealth of the people of the pair is a power of 3. A group of people wants to eat at the restaurant. Mathematically, if there are two people of wealth a and b, it forms a valid pair if �+�=3� for some positive integer k. They want to know how many possible pairs would be allowed entry.

**Task**

Given the individual wealth of the people, find the number of valid pairs.

*Notes*

* One person can be in multiple valid pairs.
* A pair of person X and Y is the same as a pair of person Y and X.

**Example**

*Assumptions*

* *N = 3*
* *wealth = [1, 2, 3]*

**Output**

1

1. **Problem** : You are given a string which comprises of lower case alphabets (a-z), upper case alphabets (A-Z), numbers, (0-9) and special characters like !,-.; etc.

You are supposed to find out **which character occurs the maximum number of times and the number of its occurrence,** in the given string. If two characters occur equal number of times, you have to output the character with the lower [ASCII value.](http://www.ascii-code.com/)

For example, if your string was: **aaaaAAAA**, your output would be: **A 4**, because **A has lower ASCII value than a.**

**Input format:**   
The input will contain a string.

**Output format:**   
You've to output two things which will be separated by a space:   
i) The character which occurs the maximum number of times.   
ii) The number of its occurrence.

**Constraints:**   
The maximum length of the string can be **1000.**

**Sample Input**

Pulkit is a dog!!!!!!!!!!!!

**Sample Output**

! 12

1. Write a Java Program to find out a non-repeated character from the Given string Like “JAVA IS GREAT AND JAVA IS GOOD” by using Streams.
2. Java's System.out.printf function can be used to print formatted output. The purpose of this exercise is to test your understanding of formatting output using printf.

To get you started, a portion of the solution is provided for you in the editor; you must format and print the input to complete the solution.

**Input Format**

Every line of input will contain a String followed by an integer.

Each String will have a maximum of  alphabetic characters, and each integer will be in the inclusive range from  to .

Output Format

In each line of output there should be two columns:

The first column contains the String and is left justified using exactly  characters.

The second column contains the integer, expressed in exactly  digits; if the original input has less than three digits, you must pad your output's leading digits with zeroes.

Sample Input

java 100

cpp 65

python 50

Sample Output

================================

java           100

cpp            065

python         050

================================

1. Given an integer, , print its first  multiples. Each multiple  (where ) should be printed on a new line in the form: N x i = result.

Input Format

A single integer, .

Constraints

Output Format

Print  lines of output; each line  (where ) contains the  of  in the form:

N x i = result.

Sample Input

2

Sample Output

2 x 1 = 2

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

2 x 5 = 10

2 x 6 = 12

2 x 7 = 14

2 x 8 = 16

2 x 9 = 18

2 x 10 = 20

1. You are given an integer , you have to convert it into a string.   If your code successfully converts  into a string  the code will print "Good job". Otherwise it will print "Wrong answer".   can range between  to  inclusive.

Sample Input 0

100

Sample Output 0

Good job

1. Using Regex, we can easily match or search for patterns in a text. Before searching for a pattern, we have to specify one using some well-defined syntax.  In this problem, you are given a pattern. You have to check whether the syntax of the given pattern is valid.

Note: In this problem, a regex is only valid if you can compile it using the Pattern.compile method.

Input Format

The first line of input contains an integer , denoting the number of test cases. The next  lines contain  a string of any printable characters representing the pattern of a regex.

Output Format

For each test case, print Valid if the syntax of the given pattern is correct. Otherwise, print Invalid. Do not print the quotes.

Sample Input

3

([A-Z])(.+)

[AZ[a-z](a-z)

batcatpat(nat

Sample Output

Valid

Invalid

Invalid

1. Two strings,  and , are called anagrams if they contain all the same characters in the same frequencies. For this challenge, the test is not case-sensitive. For example, the anagrams of CAT are CAT, ACT, tac, TCA, aTC, and CtA.

Function Description

Complete the isAnagram function in the editor.

isAnagram has the following parameters:

• string a: the first string

• string b: the second string

Returns

• boolean: If  and  are case-insensitive anagrams, return true. Otherwise, return false.

Input Format

The first line contains a string .

The second line contains a string .

Constraints

• Strings  and  consist of English alphabetic characters.

• The comparison should NOT be case sensitive.

Sample Input 0

anagram

margana

Sample Output 0

Anagrams

1. Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.

2. Open brackets must be closed in the correct order.

3. Every close bracket has a corresponding open bracket of the same type.

Example 1:

Input: s = "()"

Output: true

Question 1: Temperature Converter

UI :

A screenshot of a computer

Description automatically generated with medium confidence

**Functionality Requirements**

* It has 2 input number fields. The first is for a Celsius value, and the second is for a Fahrenheit value.
* Initially, both fields are empty.
* As a value is typed into the Celsius field, convert it to Fahrenheit and show it in the Fahrenheit field. Use the formula F = C\*9/5 + 32 for conversion. In case of decimals, show up to 1 decimal value.
* As a value is typed into the Fahrenheit field, convert it to Celsius and show it in the Celsius field. Use the formula C = (F − 32) × 5/9 for conversion. In case of decimals, show up to 1 decimal value.

**Testing Requirements**

* The Celsius input should have the data-test-id attribute 'celsius-input'.
* The Fahrenheit input should have the data-test-id attribute 'fahrenheit-input'.

Question 2: Weather Details component

UI :

A screenshot of a weather details

Description automatically generated with low confidence

**Functionality Requirements**

* An array of objects is passed as a prop to the component, where each object is a weather record for a single city. The object has 4 properties:

1. name: The name of the city. [STRING]
2. temperature: The temperature in the city. [STRING]
3. wind: The wind in the city. [STRING]
4. humidity: The humidity in the cit.y [STRING]

* There is an input field for the city name where the user can type the name of a city to search the weather data for. (The city name is case-insensitive.)
* If data exists for the typed input, render the weather details <div> as below, inside <div data-test-id="weather-details">.

1. <span data-test-id="output-temperature">{temperature}</span>, where {temperature} is the value from the weather record.
2. <div data-test-id="output-wind">Wind: {wind}</div>, where {wind} is the value from the weather record.
3. <div data-test-id="output-humidity">Humidity: {humidity}</div>, where {humidity} is the value from the weather record.

* If no data exists for the typed input, do not render the weather details <div>, but instead render <div data-test-id="no-results">No Results Found</div>.
* At component render, since nothing is typed, do not render above 2 divs.

**Testing Requirements**

* The city name input should have the data-test-id attribute 'app-input'.
* The <div> containing weather details should have the data-test-id attribute 'weather-details'.
* The <span> containing the temperature should have the data-test-id attribute 'output-temperature'.
* The <div> containing the wind information should have the data-test-id attribute 'output-wind'.
* The <div> containing the humidity information should have the data-test-id attribute 'output-humidity'.
* The 'No Results Found' <div> should have the data-test-id attribute 'no-results'.

Question 3: Survey List

UI:

A screenshot of a phone

Description automatically generated with medium confidence

**Functionality Requirements**

There are 2 components in the app:

* Filters component: A reusable component that is used to define the filters for the final survey list to be rendered. It accepts the filter type and the list of filter values as input, and outputs the selected filter based on the criteria mentioned below.
* SurveyList component: This component is used to render a list of surveys.

The app should have the following functionalities:

* The app should render the list of Survey objects. - The interface for an object is defined in the file src/types/Survey.ts having the following structure:

  interface Survey {

    title: string;

    category: string; // Possible values - 'Workplace', 'Development' or 'Hardware'

    status: string; // // Possible values - 'Active', or 'Completed'

    label: string;

  }

* In the left pane, we have 2 Filters component instances -
* The first filter instance filters the surveys based on the status property. Clicking on a filter should render the survey objects for which the status value matches the filter value. In case All is clicked, this filter should have no effect and should not filter out any surveys.
* The second filter instance filters the surveys based on the category property. Clicking on a filter should render the survey objects for which the category value matches the filter value. - The first click selects a filter, second consecutive click unselects the filter.
* If both the above mentioned filters have a value selected, the surveys should be filtered based on the combination of both the filters. For eg: If the first filter has Active selected and the second filter has Completed selected, show all Active surveys whose category is Completed in the right pane.
* In the right pane, we have the SurveyList component that renders the filtered survey items in a list item <li> inside the list <ul data-test-id="survey-list"></ul>.
* Initially, all surveys should be rendered in the SurveyList component.

**Testing Requirements**

* The <ul> containing status filters should have the data-test-id attribute status-list.
* The <ul> containing category filters should have the data-test-id attribute category-list.
* The output <ul> should have the data-test-id attribute survey-list.

Question 4: Weather Finder API

UI:

A screenshot of a computer

Description automatically generated with medium confidence

Functionality requirements:

The component must have the following functionalities:

* The input should initially be empty. The user can type a city name into this input box to search for weather details for this city.
* Clicking on the Search button should make an API GET call to the URL https://jsonmock.hackerrank.com/api/weather?name=<name> using the Angular HttpClient module. Here, <name> is the city name entered into the text box. For example, for the value Dallas, the API hit has to be https://jsonmock.hackerrank.com/api/weather?name=Dallas. You will always get data for cities Dallas and Oakland.
* The response contains a data field, where data is an array of objects, and each object is a weather record. We only need to use the first record from the array for rendering in this challenge. The sample format of the data field is given below:

  "data": [

    {

      "name": "Dallas",

      "weather": "12 degree", // Format is always "<value> degree"

      "status": [

        "Wind: 2Kmph", // String

        "Humidity: 5%" // String

      ]

    }

  ]

* The weather details should be rendered inside <div data-test-id="weather-details"></div>. This div should not be rendered initially since no API has been hit yet.
* Each weather record contains a weather field. Retrieve the value and display in the following element - <span data-test-id="result-temperature"></span>.
* If value in weather field is less than 20, render cold weather icon by rendering <i data-test-id="icon-cold"></i>. If the value is greater than or equal to 20, render sunny weather icon by rendering <i data-test-id="icon-sunny"></i>.
* Each weather record contains a status field which is an array of strings.
* The first string denotes the wind value and the second string denotes the humidity value.

1. Render wind value in
2. Render humidity value in

* If no records are returned for any city by the API, you must render <div data-test-id="no-result">No Results Found</div> instead, and this element must be visible only when the data field is an empty array. This div should not be rendered initially since no API has been hit yet.
* Please note that the input field accepts only text. Test cases take care of calling the API with valid input, so writing input validation is not required.
* For testing purposes, please use the following cities and their corresponding weather conditions:

  Dallas - Cold

  Oakland - Sunny

Testing requirements:

* The input should have the data-test-id attribute app-input.
* The Search button should have the data-test-id attribute submit-button.
* The weather details should have the data-test-id attribute weather-details.
* The sunny icon should have the data-test-id attribute icon-sunny.
* The cold icon should have the data-test-id attribute icon-cold.
* The span showing temperature should have the data-test-id attribute result-temperature.
* The div showing wind information should have the data-test-id attribute result-wind.
* The div showing wind information should have the data-test-id attribute result-humidity.
* The No Results Found div should have the data-test-id attribute no-result.

Question 5: User List

UI:

A screenshot of a computer

Description automatically generated

Functionality requirements:

There are 2 components in the app:

* DataForm component: This component is used to add a new item of type Song or Book to the list.
* DataList component: A reusable component that is used to render the list of Songs and the list of Books. Accepts the appropriate List and the dataType (One of "Song" or "Book") as input.

The app should have the following functionalities:

* The user can add items to the book list or the song list from the same form. Adding an item in the form at the top should add it to the respective list below.
* There are 3 required input fields - name, genre, creator, and a type input field having 2 options to choose if the current item being added is a book or a song.
* For this challenge, you can assume that each item is uniquely identified by its name. Tests take care of testing with unique names only.
* On choosing Song, render an extra input field totalTime. Initially, all fields should be empty.
* Clicking on Add button should add the item to the respective list and clear all the input fields.
* The DataList component renders each of book list and song list in a table having columns name, genre, creator of each item followed by a delete button. Clicking on the delete button should delete the respective item from the list. Song list has an extra column of Time to render Total Time information for the item.
* Book item should be added to <table data-test-id="book-table"> as a <tr>.
* Song item should be added to <table data-test-id="song-table"> as a <tr>.
* The interface for an item is defined in the file src/types/Item.ts having the following structure:

  interface Item {

    name: string;

    genre: string;

    creator: string;

    type: string;

    totalTime?: number;

  }

Testing requirements:

* The input field for name has data-test-id attribute app-input-name.
* The input field for genre has data-test-id attribute app-input-genre.
* The input field for creator has data-test-id attribute app-input-creator.
* The input field for total time has data-test-id attribute app-input-time.
* The input field for type Book has data-test-id attribute app-input-book-type.
* The input field for type Song has data-test-id attribute app-input-song-type.
* The Add button has data-test-id attribute add-button.
* The book table has data-test-id attribute book-table.
* The song table has data-test-id attribute song-table.
* Rows in a single table has data-test-id attribute list-item-0, list-item-1 and so on.
* The Cell having name has data-test-id attribute item-name.
* The Cell having name has data-test-id attribute item-name.
* The Cell having creator has data-test-id attribute item-creator.
* The Cell having total time has data-test-id attribute item-time.
* The Cell having delete button has data-test-id attribute item-delete.

Question 6: Length Converter

UI:

A screenshot of a computer

Description automatically generated with medium confidence

Functionality requirements:

The component should have the following functionalities:

* It has 2 input set. Each input set has the following:
* An input field to type the number which has a label defining unit of measurement selected.
* A dropdown that has 3 options - Kilometre, Metre, Centimetre. The array containing the objects of all possible values in the dropdown is below. Use the id property of each object to pass the value to each option as that is used for testing.
* [
* {
* id: 0,
* label: 'Kilometre',
* unit: 'km'
* },
* {
* id: 1,
* label: 'Metre',
* unit: 'm'
* },
* {
* id: 2,
* label: 'Centimetre',
* unit: 'cm'
* }
* ]
* Initially, both the input fields are empty. The first dropdown should have Kilometre selected on app load and the second dropdown should have Metre selected on app load. Therefore, first input should have the label as km and second input should have the label as m.
* As and when a value is typed in either input field, convert it to unit selected in the other input set and update the value of other input field.
* Please note, when input fields have values, changing unit from dropdown should update the value of respective input field.
* Example - first dropdown has Kilometres selected and second dropdown has Metres selected. Any value in typed in first input field should render the converted Metre value in second input field.
* Use the following conversions:

  1 Kilometre = 1000 Metres

  1 Metre = 100 Centimetres

Testing requirements:

The following data-test-id attributes are required in the component for the tests to pass:

* In the first input set -
* Input field has data-test-id attribute app-input1.
* Label has data-test-id attribute app-label1.
* Select has data-test-id attribute app-select1.
* In the second input set -
* Input field has data-test-id attribute app-input2.
* Label has data-test-id attribute app-label2.
* Select has data-test-id attribute app-select2.

Question 7: Football competitions API

UI :

A screenshot of a sports website

Description automatically generated with low confidence

Functionality requirements:

The main aim is to get a paginated list of football competitions and render their details in a list. The component must have the following functionalities:

* The component must get competitions by making an API GET call to URL https://jsonmock.hackerrank.com/api/football\_competitions?page=<pageNumber> using the Angular HttpClient module. Here, <pageNumber> is the page number we want to get the data for.
* The response of the GET call will contain a total\_pages field that denotes the number of pages of results available and a data field that is an array of competition records for the requested page. The sample format of the response is:

  {

      "page": "1",

      "per\_page": 2,

      "total": 2,

      "`total\_pages`": 1,

      "data": [

        {

          "name": "English Premier League",

          "country": "England",

          "year": 2016,

          "winner": "Chelsea",

          "runnerup": "Tottenham Hotspur"

        },

        {

          "name": "La Liga",

          "country": "Spain",

          "year": 2011,

          "winner": "Real Madrid",

          "runnerup": "FC Barcelona"

        }

    ]

  }

* On component mount, make a GET call to get the data for page 1 (i.e., API GET call to URL https://jsonmock.hackerrank.com/api/football\_competitions?page=1.
* Retrieve total\_pages from the response and render pagination buttons corresponding to each page starting from 1 to total\_pages. Each button must be rendered as <button>{k}</button>, where {k} is the page number the button corresponds to, for example <button>1</button>, <button>2</button>, and so on until <button>{total\_pages}</button>.
* All the buttons must be rendered in the section <section data-test-id="page-number-buttons"></section>.
* Clicking on a page button must get records for the corresponding page number and render them. For example, clicking on button 3 must make an API GET call to URL https://jsonmock.hackerrank.com/api/football\_competitions?page=3, get the data, and render it.
* For the competitions returned by the API, you need to render the list <ul data-test-id="football-competitions"></ul>. This list should have a single <li> list item for each object in the array. The value of each <li> element should be <li>Competition {name} won by {winner} in year {year}</li> where {name}, {winner} and {year} are values retrieved from the corresponding competition object.
* For example, in the above data example, there are 2 competition objects in the array, so there will be 2 <li> elements inside the <ul> element:

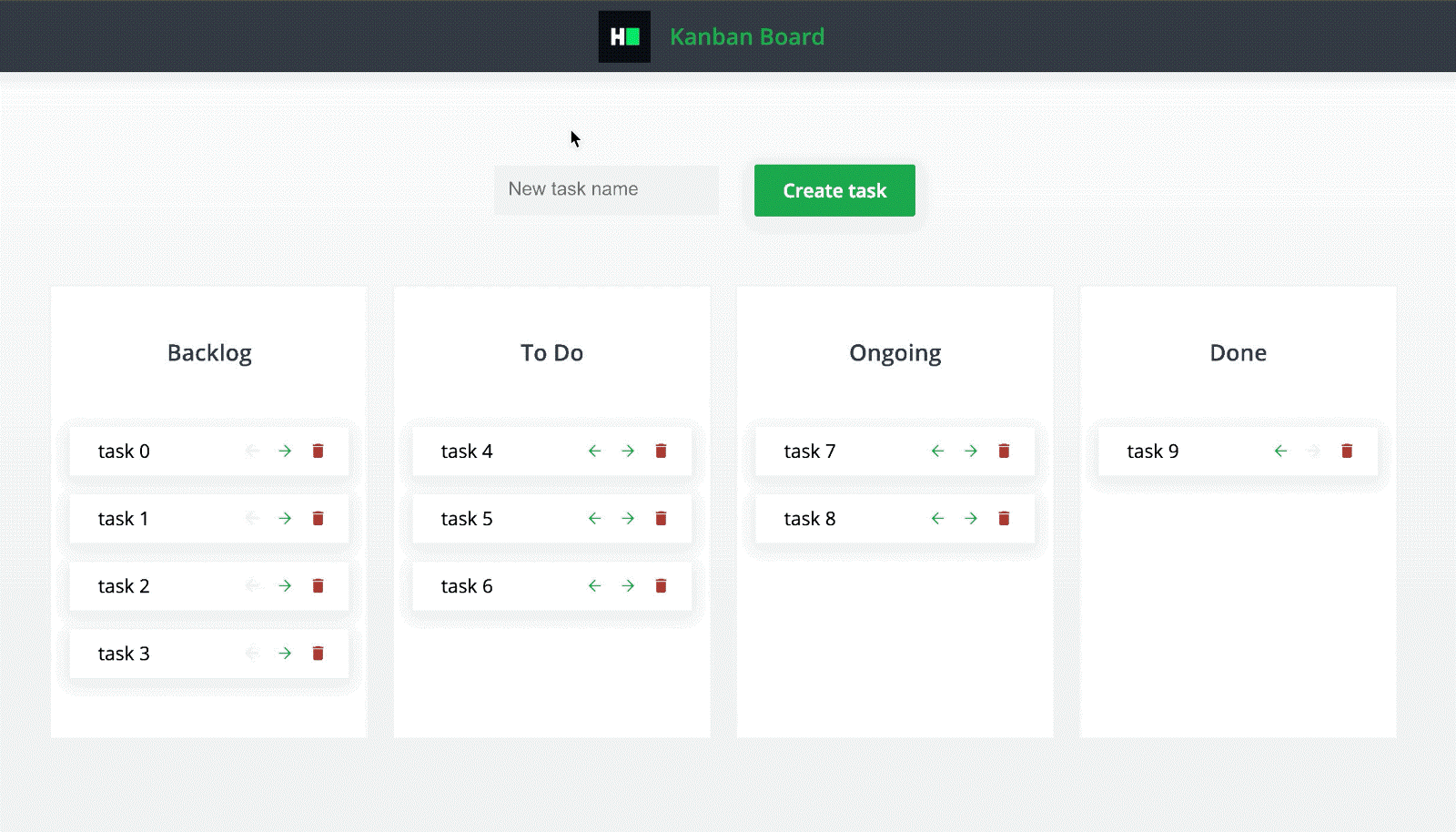
1. <li>Competition English Premier League won by Chelsea in year 2016</li>
2. <li>Competition La Liga won by Real Madrid in year 2011</li>

Testing requirements:

* The <section> containing all the buttons should have the data-test-id attribute page-number-buttons.
* The <ul> should have data-test-id attribute football-competitions.

Question 8: Kanban Board

UI :



Functionality requirements:

* The board contains 4 stages of tasks in sequence - 'Backlog', 'To Do', 'Ongoing' and 'Done'.
* The 'New Task Name' input should initially be empty. The user can type a task name in this input box and clicking on 'Create task' button should add a new task with this task name. This newly created task should be added to the Backlog stage (the first stage). Post this, clear the input field.
* If 'Create Task' button is clicked with input being empty, nothing should happen.
* In every individual stage, the tasks are rendered as a list <ul> where each task is a single list item <li> which displays the name of the task.
* Each task list item has 3 icon buttons at the right -

1. Back button - This moves the task to the previous stage in sequence, if any. This button is disabled if the task is in the first stage.
2. Forward button - This moves the task to the next stage in sequence, if any. This button is disabled if the task is in the last stage.
3. Delete button - This removes the task from the board.

* Each task has 2 properties -

1. name - name of task. This is the unique identification for every task. [STRING]
2. stage - stage of task [NUMBER] (0 represents Backlog stage, 1 represents To Do stage, 2 represents Ongoing stage, 3 represents Done stage).

Testing requirements:

* Input should have the data-test-id attribute 'create-task-input'.
* 'Create task' button should have the data-test-id attribute 'create-task-button'.
* <ul> for 'Backlog' stage should have the data-test-id attribute 'stage-0'.
* <ul> for 'To Do' stage should have the data-test-id attribute 'stage-1'.
* <ul> for 'Ongoing' stage should have the data-test-id attribute 'stage-2'.
* <ul> for 'Done' stage should have the data-test-id attribute 'stage-3'.
* Every single <li> task should have below:

1. The <span> containing the name should have data-test-id attribute 'TASK\_NAME-name' where TASK\_NAME is the name of the task joined by hyphen symbol. For example, for the task named "task 0", it should be 'task-0-name'. For the task named "abc", it should be 'abc-name'.
2. The back button should have data-test-id attribute 'TASK\_NAME-back' where TASK\_NAME is the name of the task joined by hyphen symbol. For example, for the task named "task 0", it should be 'task-0-back'. For the task named "abc", it should be 'abc-back'.
3. The forward button should have data-test-id attribute 'TASK\_NAME-forward' where TASK\_NAME is the name of the task joined by hyphen symbol. For example, for the task named "task 0", it should be 'task-0-forward'. For the task named "abc", it should be 'abc-forward'.
4. The delete button should have data-test-id attribute 'TASK\_NAME-delete' where TASK\_NAME is the name of the task joined by hyphen symbol. For example, for the task named "task 0", it should be 'task-0-delete'. For the task named "abc", it should be 'abc-delete'.

Question 9: Money converter

Create a Money Converter component, as shown below:

The component should have the following functionalities:

* It has 2 input number fields. The first is for a Dollar value, and the second is for a Rupee value.
* Initially, both fields are empty.
* As a value is typed into the Dollar field, convert it to Rupee and show it in the Rupee field. Use the formula D = R \* 80 for conversion. In case of decimals, show up to 1 decimal value.
* As a value is typed into the Rupee field, convert it to Dollar and show it in the Dollar field. Use the formula R = D / 80 for conversion. In case of decimals, show up to 1 decimal value.

Question 10: Country-Capital Component

Create a Country Details component, as shown below:

The component must have the following functionalities:

* An array of objects is passed as a prop to the component, where each object is a capital record for a single city. The object has 2 properties:
* country Name: The name of the country. [STRING]
* capital Name: The name of the city. [STRING]
* There is an input field for the country name where the user can type the name of a country to search the weather data for. (The city name is case-insensitive.)
* If data exists for the typed input, render the capital name.
* If no data exists for the typed input, do not render the country details <div>, but instead render message: *No Results Found.*
* At component render, since nothing is typed, do not render the above two <div> elements.

Question 11: Volume converter

Create a Volume Converter component, as shown below:

The component should have the following functionalities:

It has 2 input set. Each input set has the following:

An input field to type the number which has a label defining unit of measurement selected.

A dropdown that has 2 options - Litres, Millilitres. The array containing the objects of all possible values in the dropdown is below. Use the 'id' property of each object to pass the value to each option.

[   
   {   
      id: 0,   
      label: 'Litres',   
      unit: 'L'   
    },   
    {   
      id: 1,   
      label: 'Millilitres',   
      unit: 'ml'   
    }

]

Initially, both the input fields are empty. The first dropdown should have ‘Litres’ selected on app load and the second dropdown should have ‘Millilitres’ selected on app load. Therefore, first input should have the label as L and second input should have the label as 'ml'.

As and when a value is typed in either input field, convert it to unit selected in the other input set and update the value of other input field.

Please note, when input fields have values, changing unit from dropdown should update the value of respective input field.

Use the following conversions:

1 litre = 1000 millilitres

**JavaScript Questions**

Question 12: Reverse words in a given string

Given a String S, reverse the string without reversing its individual words. Words are separated by dots.

**Example 1:**

**Input:**

S = i.like.this.program.very.much

**Output:** much.very.program.this.like.i

**Explanation:** After reversing the whole

string(not individual words), the input

string becomes

much.very.program.this.like.i

**Example 2:**

**Input:**

S = pqr.mno

**Output:** mno.pqr

**Explanation:** After reversing the whole

string , the input string becomes

mno.pqr

**Your Task:**   
You dont need to read input or print anything. Complete the function**reverseWords()** which takes string S as input parameter and returns a string containing the words in reversed order. Each word in the returning string should also be separated by '.'

Question 12: Angaram

Given two strings **a**and **b**consisting of lowercase characters. The task is to check whether two given strings are an anagram of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. For example, act and tac are an anagram of each other.

**Note:-**

* If the strings are anagrams you have to **return True or else return False**
* **|s|**represents the length of string s.

**Example 1:**

**Input:**a = geeksforgeeks, b = forgeeksgeeks

**Output:** YES

**Explanation:** Both the string have same characters with

        same frequency. So, both are anagrams.

**Example 2:**

**Input:**a = allergy, b = allergic

**Output:** NO

**Explanation:** Characters in both the strings are

        not same, so they are not anagrams.

**Your Task:**   
You don't need to read input or print anything. Your task is to complete the function **isAnagram()** which takes the string **a** and string **b** as input parameter and check if the two strings are an anagram of each other. The function returns true if the strings are anagram else it returns false.

Question 13: Find the frequencies of all duplicates elements in the array

Given an array of integers with duplicate elements in it, the task is to find the duplicate elements in the array and their frequencies.

**Examples:**

***Input:****arr[] = {2, 3, 4, 5, 4, 6, 4, 7, 4, 5, 6, 6}*   
***Output:****Below is the frequency of repeated elements –*   
*4 –> 4*   
*5 –> 2*   
*6 –> 3*   
***Input:****arr[] = {4, 4, 5, 5, 6}*   
***Output:****Below is the frequency of repeated elements –*   
*4 –> 2*   
*5 –> 2*

Question 14: Coin change

Given an integer array **coins[ ] of size N**representing different denominations of currency and an integer **sum**, find the number of ways you can make **sum** by using different combinations from coins[ ].     
Note: Assume that you have an infinite supply of each type of coin.

**Example 1:**

**Input:**

sum = 4 ,

N = 3

coins[] = {1,2,3}

**Output:** 4

**Explanation**: Four Possible ways are:

{1,1,1,1},{1,1,2},{2,2},{1,3}.

**Example 2:**

**Input**:

Sum = 10 ,

N = 4

coins[] ={2,5,3,6}

**Output:** 5

**Explanation**: Five Possible ways are:

{2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5}

and {5,5}.

**Your Task:**   
You don't need to read input or print anything. Your task is to complete the function **count()**which accepts an array coins**[ ] its size N and sum** as input parameters and returns the number of ways to make change for given sum of money.

Question 15: Edit Distance

Given two strings **s** and **t.**Return the minimum number of operations required to convert **s**to **t**.   
The possible operations are permitted:

1. Insert a character at any position of the string.
2. Remove any character from the string.
3. Replace any character from the string with any other character.

**Example 1:**

**Input:**

s = "geek", t = "gesek"

**Output:** 1

**Explanation:** One operation is required

inserting 's' between two 'e's of s.

**Example 2:**

**Input :**

s = "gfg", t = "gfg"

**Output:**

0

**Explanation:** Both strings are same.

**Your Task:**   
You don't need to read or print anything. Your task is to complete the function **editDistance()**which takes strings s and t as input parameters and returns the minimum number of operation to convert the string **s** to string**t**.