Python Interview

Question 1: (Logical, Score - 100)

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. For each match of team B, compute the total number of matches of team A where team A has scored *less than or equal to* the number of goals scored by team B in that match.

Example

teamA = [1, 2, 3]

teamB = [2, 4]

Team A has played three matches and has scored *teamA = [1, 2, 3]* goals in each match respectively. Team B has played two matches and has scored *teamB = [2, 4]* goals in each match respectively. For *2* goals scored by team B in its first match, team A has *2* matches with scores *1* and *2*. For *4* goals scored by team B in its second match, team A has *3* matches with scores *1*, *2* and *3*. Hence, the answer is *[2, 3]*.

Function Description

Complete the function *counts* in the editor below.

counts has the following parameter(s):

int teamA[n]: first array of positive integers

int teamB[m]: second array of positive integers

Return

int[m]: an array of *m* positive integers, one *for each teamB[i]* representing the total number of elements from *teamA[j]* satisfying *teamA[j] ≤ teamB[i]* where *0 ≤ j < n* and *0 ≤ i < m,* in the given order.

Constraints

- $2 \le n, m \le 10^5$
- $1 \le teamA[j] \le 10^9$, where $0 \le j < n$.
- $1 \le teamB[i] \le 10^9$, where $0 \le i < m$.

▼ Sample Case 0

Sample Input 0

```
STDIN       Function
-----       --------
4       →   teamA[] size n = 4
1       →   teamA = [1, 4, 2, 4]
4
2
4
2       →   teamB[] size m = 2
3       →   teamB = [3, 5]
5
```

Sample Output 0

```
2
4
```

Explanation 0

Given values are *n = 4, teamA = [1, 4, 2, 4], m = 2,* and *teamB = [3, 5].*

1. For *teamB[0] = 3*, we have *2* elements in *teamA* (*teamA[0] = 1* and *teamA[2] = 2*) that are *≤ teamB[0].*
2. For *teamB[1] = 5*, we have *4* elements in *teamA* (*teamA[0] = 1, teamA[1] = 4, teamA[2] = 2,* and *teamA[3] = 4*) that are *≤ teamB[1].*

Thus, the function returns the array *[2, 4]* as the answer.

▼ Sample Case 1

Sample Input 1

```
STDIN       Function
-----       --------
5       →   teamA[] size n = 5
2       →   teamA = [2, 10, 5, 4, 8]
10
5
4
8
4       →   teamB[] size m = 4
3       →   teamB = [3, 1, 7, 8]
1
7
8
```

Sample Output 1

```
1
0
3
4
```

Explanation 1

Given values are *n = 5*, *teamA = [2, 10, 5, 4, 8]*, *m = 4*, and *teamB = [3, 1, 7, 8]*.

1. For *teamB[0] = 3*, we have *1* element in *teamA* (*teamA[0] = 2*) that is ≤ *teamB[0]*.
2. For *teamB[1] = 1*, there are *0* elements in *teamA* that are ≤ *teamB[1]*.
3. For *teamB[2] = 7*, we have *3* elements in *teamA* (*teamA[0] = 2*, *teamA[2] = 5*, and *teamA[3] = 4*) that are ≤ *teamB[2]*.
4. For *teamB[3] = 8*, we have *4* elements in *teamA* (*teamA[0] = 2*, *teamA[2] = 5*, *teamA[3] = 4*, and *teamA[4] = 8*) that are ≤ *teamB[3]*.

Thus, the function returns the array *[1, 0, 3, 4]* as the answer.

Question 2: (Database, Score - 100)

A pizza company is taking orders from customers, and each pizza ordered is added to their database as a separate order. Each order has an associated status, "*CREATED or SUBMITTED or DELIVERED*". An order's Final_Status is calculated based on *status* as follows:

1. When all orders for a customer have a status of *DELIVERED*, that customer's order has a Final_Status of *COMPLETED*.
2. If a customer has some orders that are not *DELIVERED* and some orders that are *DELIVERED*, the Final_Status is *IN PROGRESS*.
3. If at least one of a customer's orders is *SUBMITTED* and none is *DELIVERED*, the Final_Status is *AWAITING PROGRESS*.
4. Otherwise, the Final_Status is *AWAITING SUBMISSION*.

Write a query to report the *customer_name* and *Final_Status* of each customer's order. Order the results by customer name.

Table definitions and a data sample are given below.

Schema

Table: **Customer_Order**

column name	column type
customer_name	varchar2(50)

order_id	varchar2(10)
status	varchar2(50)

Sample Data Tables

Table: **Customer_Order**

Customer_name	order_id	status
John	J1	DELIVERED
John	J2	DELIVERED
David	D1	DELIVERED
David	D3	CREATED
Smith	S1	SUBMITTED
KRISH	K1	CREATED

When all orders are in DELIVERED status then the Final_Status is "COMPLETED". When one or more orders for a customer are DELIVERED and one or more orders are CREATED or SUBMITTED, then the Final_Status is "IN PROGRESS". When one or more orders for a customer are SUBMITTED and none are DELIVERED, then the Final_Status is "AWAITING PROGRESS". Otherwise, the Final_Status is "AWAITING SUBMISSION".

Order priority is DELIVERED , SUBMITTED, CREATED.

The results are:

customer_name	Final_Status
David	IN PROGRESS
John	COMPLETED
KRISH	AWAITING SUBMISSION
Smith	AWAITING PROGRESS

Question 3: (REST API, Score - 100)

Query a REST API to get a list of articles. Given an integer, *limit*, return the top *limit* article names ordered decreasing by comment count, then decreasing alphabetically for those that have the same comment counts.

To access the collection of comments, make an HTTP GET request to:
https://jsonmock.hackerrank.com/api/articles?page=<pageNumber>
where <pageNumber> is an integer where *1 <= pageNumber <= total_pages. total_pages* is one of the fields in the JSON data.

The response is a JSON object with the following 5 fields:

- page: The current page of the results
- per_page: The maximum number of records returned per page.
- total: The total number of records on all pages of the result.
- total_pages: The total number of pages with results.
- data: An array of objects containing records returned on the requested page

Each record in *data* has the following schema.

- title: the title of the article, may be null
- url: the URL of the article
- author: the username of the author of the article
- num_comments: the number of comments the article has, may be null (no comments)
- story_id: identifier of the story related to the article, may be null
- story_title: the title of the story related to the article, may be null
- story_url: the URL of the story related to the article, may be null
- parent_id: identifier of the parent of the article, may be null
- created_at: the date and time the record was created

First get the article name.

- If the *title* field is not null, use *title*.
- Otherwise, if the *story_title* field is not null, use *story_title.*
- If both fields are null, ignore the article.

Sort the titles decreasing by comment count, then decreasing alphabetically by article name if there is a tie in comments count. Return a list of the top *limit* names.

Function Description

Complete the function *topArticles* in the editor below.

topArticles has the following parameter(s):

int limit: the number of articles to return

Returns

string[k]: the names of articles

▼ Input Format For Custom Testing

In the first line, there is an integer *limit*.

▼ Sample Case 0

Sample Input For Custom Testing

```
2
```

Sample Output

```
UK votes to leave EU
F.C.C. Repeals Net Neutrality Rules
```

Explanation

The limit value is 2 so return the names of the top two articles based on the number of comments. Those top articles are:

1. `title`: F.C.C. Repeals Net Neutrality Rules, `story_title`: null, `num_comments`: 1397
2. `title`: UK votes to leave EU, `story_title`: null, `num_comments`: 2531

Their names are their titles because they are not null. The second of these articles has more comments, so it comes first. There is not a tie for comment count so there is no change from the secondary sort.

Kindly have an AWS account created to perform Question 4 and 5

Question 4: (AWS Lambda, Score - 100)

Implement an AWS lambda function for simple calculator. The operation is passed as path parameter and the arguments are passed as query strings, for example *your_api_gateway_to_lambda/calculator/add?a=3&b=5* or *your_api_gateway_to_lambda/calculator/multiply?a=10&b=3*. Include addition, subtraction, multiplication, division as the only operation for the lambda function. Configure appropriate API gateway for the lambda function. Additionally write simple test cases to test all operations implemented in AWS Lambda. Provide screenshots for all operations performed.

Question 5: (AWS S3 upload, Score - 100)

Implement/host a static website on S3 using python boto3.

Instructions: Upload a static website contained in a local directory to a bucket in S3.