Q-1 Table: Employee

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| empId | int |

| name | varchar |

| supervisor | int |

| salary | int |

+-------------+---------+

empId is the column with unique values for this table.

Each row of this table indicates the name and the ID of an employee in addition to their salary and the id of their manager.

Table: Bonus

+-------------+------+

| Column Name | Type |

+-------------+------+

| empId | int |

| bonus | int |

+-------------+------+

empId is the column of unique values for this table.

empId is a foreign key (reference column) to empId from the Employee table.

Each row of this table contains the id of an employee and their respective bonus.

Write a solution to report the name and bonus amount of each employee with a bonus less than 1000.

Return the result table in any order.

The result format is in the following example.

**Select E.name , B.bonus from Employee E left join Bonus B on E.empId=B.empId**

**Where B.bonus < 1000**

Example :

Input:

Employee table:

+-------+--------+------------+--------+

| empId | name | supervisor | salary |

+-------+--------+------------+--------+

| 3 | Brad | null | 4000 |

| 1 | John | 3 | 1000 |

| 2 | Dan | 3 | 2000 |

| 4 | Thomas | 3 | 4000 |

+-------+--------+------------+--------+

Bonus table:

+-------+-------+

| empId | bonus |

+-------+-------+

| 2 | 500 |

| 4 | 2000 |

+-------+-------+

Output:

+------+-------+

| name | bonus |

+------+-------+

| Brad | null |

| John | null |

| Dan | 500 |

+------+-------+

Q-2: Write a MongoDB query to find all documents in the products collection where the product price is greater than $100 and the product name starts with the letter "A".

products:  
  - \_id: 1  
    name: Apple iPhone 13 Pro  
    price: 1099  
    quantity: 10  
  - \_id: 2  
    name: AirPods Pro  
    price: 249  
    quantity: 20  
  - \_id: 3  
    name: iPad Pro 12.9  
    price: 1099  
    quantity: 5

Q-3 Given a string, find the longest substring that does not contain any repeated characters.

Example:

*console.log(longestSubstringWithoutRepeatingCharacters("abcabcbb")); Output 🡪 3*

*console.log(longestSubstringWithoutRepeatingCharacters("bbbbb")); Output 🡪 1*

***var name="eabcabcbbd"***

***var repatedChar=[]***

***for(let i=0;i< name.length;i++)***

***{***

***let count=1;***

***for(let j=0;j<name.length;j++){***

***if(i!=j){***

***if(name[i]==name[j]){***

***count=count+1;***

***if(count >= 2){***

***if(!repatedChar.includes(name[i])){***

***repatedChar.push(name[i])***

***}***

***}***

***}***

***}***

***}***

***}***

***console.log(repatedChar)***

Q- 4 : Given an array of positive integers arr, return the sum of all possible odd-length subarrays of arr.

A subarray is a contiguous subsequence of the array.

Example 1:

Input: arr = [1,4,2,5,3]

Output: 58

Input: arr = [1,2]

Output: 3

Example 3:

Input: arr = [10,11,12]

Output: 66

**var arr = [1,4,2,5,3];**

**var sum=0;**

**for(let i=0;i<arr.length;i++){**

**for(let j=i+1;j<arr.length;j++){**

**sum=sum+arr[i]+arr[j]**

**}**

**}**

**console.log('sum=',sum);**