Lab 1

SQL :

Problem 1 :   
Query all attributes of every Japanese city in the **CITY** table. The COUNTRYCODE for Japan is JPN.   
Solution :   
<https://www.hackerrank.com/rest/contests/master/challenges/japanese-cities-attributes/hackers/hoangkhanhgk7/download_solution>

Problem 2 :

Query the names of all the Japanese cities in the **CITY** table. The COUNTRYCODE for Japan is JPN.

<https://www.hackerrank.com/rest/contests/master/challenges/japanese-cities-name/hackers/hoangkhanhgk7/download_solution>

Problem 3 :

Query a list of *CITY* and *STATE* from the **STATION** table.

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-1/hackers/hoangkhanhgk7/download_solution>

Problem 4 :

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.

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-3/hackers/hoangkhanhgk7/download_solution>

Problem 5 :

Let  be the number of *CITY* entries in **STATION**, and let  be the number of distinct *CITY* names in **STATION**; query the value of  from **STATION**. In other words, find the difference between the total number of *CITY* entries in the table and the number of distinct *CITY* entries in the table.

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-4/hackers/hoangkhanhgk7/download_solution>

Problem 6 :

Query the two cities in **STATION** with the shortest and longest *CITY* names, as well as their respective lengths (i.e.: number of characters in the name). If there is more than one smallest or largest city, choose the one that comes first when ordered alphabetically.

**Sample Input**

Let's say that *CITY* only has four entries: *DEF*, *ABC*, *PQRS* and *WXY*

**Sample Output**

ABC 3

PQRS 4

**Explanation**

When ordered alphabetically, the *CITY* names are listed as *ABC, DEF, PQRS,* and *WXY*, with the respective lengths  and . The longest-named city is obviously *PQRS*, but there are  options for shortest-named city; we choose *ABC*, because it comes first alphabetically.

**Note**  
**You can write two separate queries to get the desired output. It need not be a single query.**

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-5/hackers/hoangkhanhgk7/download_solution>

Problem 7 :

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

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-6/hackers/hoangkhanhgk7/download_solution>

Problem 8 :

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

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-7/hackers/hoangkhanhgk7/download_solution>

Problem 9 :

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.

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-8/hackers/hoangkhanhgk7/download_solution>

Problem 10 :

Query the list of *CITY* names from **STATION** that *do not start* with vowels. Your result cannot contain duplicates.

<https://www.hackerrank.com/rest/contests/master/challenges/weather-observation-station-9/hackers/hoangkhanhgk7/download_solution>

Python 3 :   
  
Problem 1 :

**Task**  
Read two integers and print two lines. The first line should contain integer division,  // . The second line should contain float division,  / .

You don't need to perform any rounding or formatting operations.

**Input Format**

The first line contains the first integer, . The second line contains the second integer, .

**Output Format**

Print the two lines as described above.

**Sample Input 0**

4

3

**Sample Output 0**

1

1.33333333333

<https://www.hackerrank.com/rest/contests/master/challenges/python-division/hackers/hoangkhanhgk7/download_solution>

Problem 2 :

**Task**  
Read two integers from STDIN and print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).
3. The third line contains the product of the two numbers.

**Input Format**

The first line contains the first integer, . The second line contains the second integer, .

**Constraints**

**Output Format**

Print the three lines as explained above.

**Sample Input 0**

3

2

**Sample Output 0**

5

1

6

<https://www.hackerrank.com/rest/contests/master/challenges/python-arithmetic-operators/hackers/hoangkhanhgk7/download_solution>

Problem 3 :

**Task**  
Read an integer . For all non-negative integers , print . See the sample for details.

**Input Format**

The first and only line contains the integer, .

**Constraints**

**Output Format**

Print  lines, one corresponding to each .

**Sample Input 0**

5

**Sample Output 0**

0

1

4

9

16

<https://www.hackerrank.com/rest/contests/master/challenges/python-loops/hackers/hoangkhanhgk7/download_solution>

Problem 4 :

Read an integer .

Without using any string methods, try to print the following:

Note that "" represents the values in between.

**Input Format**

The first line contains an integer .

**Output Format**

Output the answer as explained in the task.

**Sample Input 0**

3

**Sample Output 0**

123

<https://www.hackerrank.com/rest/contests/master/challenges/python-print/hackers/hoangkhanhgk7/download_solution>

Problem 5 :

Let's learn about list comprehensions! You are given three integers  and  representing the dimensions of a cuboid along with an integer . You have to print a list of all possible coordinates given by  on a 3D grid where the sum of  is not equal to . Here,

**Input Format**

Four integers  and  each on four separate lines, respectively.

**Constraints**

Print the list in lexicographic increasing order.

**Sample Input 0**

1

1

1

2

**Sample Output 0**

[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]

**Explanation 0**

**Concept**

You have already used lists in previous hacks. List comprehensions are an elegant way to build a list without having to use different for loops to append values one by one. This example might help.

**Example:** You are given two integers x and y . You need to find out the ordered pairs ( i , j ) , such that ( i + j ) is not equal to n and print them in lexicographic order.( 0 <= i <= x ) and ( 0 <= j <= y) This is the code if **we dont use list comprehensions in Python**.

python x = int ( raw\_input()) y = int ( raw\_input()) n = int ( raw\_input()) ar = [] p = 0 for i in range ( x + 1 ) : for j in range( y + 1): if i+j != n: ar.append([]) ar[p] = [ i , j ] p+=1 print ar  
Other smaller codes may also exist, but using list comprehensions is always a good option. **Code using list comprehensions:**

python x = int ( raw\_input()) y = int ( raw\_input()) n = int ( raw\_input()) print [ [ i, j] for i in range( x + 1) for j in range( y + 1) if ( ( i + j ) != n )]

**Sample Input 1**

2

2

2

2

**Sample Output 1**

[[0, 0, 0], [0, 0, 1], [0, 1, 0], [0, 1, 2], [0, 2, 1], [0, 2, 2], [1, 0, 0], [1, 0, 2], [1, 1, 1], [1, 1, 2], [1, 2, 0], [1, 2, 1], [1, 2, 2], [2, 0, 1], [2, 0, 2], [2, 1, 0], [2, 1, 1], [2, 1, 2], [2, 2, 0], [2, 2, 1], [2, 2, 2]]

<https://www.hackerrank.com/rest/contests/master/challenges/list-comprehensions/hackers/hoangkhanhgk7/download_solution>