Generate the following two result sets:

1. Query an alphabetically ordered list of all names in **OCCUPATIONS**, immediately followed by the first letter of each profession as a parenthetical (i.e.: enclosed in parentheses). For example: `AnActorName(A)`, `ADoctorName(D)`, `AProfessorName(P)`, and `ASingerName(S)`.
2. Query the number of ocurrences of each occupation in **OCCUPATIONS**. Sort the occurrences in ascending order, and output them in the following format:

```
There are a total of [occupation_cour
```

where `[occupation_count]` is the number of occurrences of an occupation in **OCCUPATIONS** and `[occupation]` is the lowercase occupation name. If more than one Occupation has the same `[occupation_count]`, they should be ordered alphabetically.

Note: There will be at least two entries in the table for each type of occupation.

Input Format

The **OCCUPATIONS** table is described as follows:

Column	*Type*
Name	*String*
Occupation	*String*

Occupation will only contain one of the following values: **Doctor**, **Professor**, **Singer** or **Actor**.

Sample Input

MS SQL Server

```
2  SELECT CONCAT (NAME, '(', LEFT (OCCUPATION, 1),
   ')') as col1
3  FROM  OCCUPATIONS
4  ORDER BY  NAME;
5
6  SELECT CONCAT ('There are a total of ', COUNT
   (OCCUPATION), ' ', lower (OCCUPATION), 's.') as
   col1
7  FROM OCCUPATIONS
8  GROUP BY OCCUPATION
9  ORDER BY COUNT(OCCUPATION), OCCUPATION;
10
```

Line: 2 Col: 1

Upload Code as File | Run Code | Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Your Output (stdout)

```
1  Aamina(D)
2  Ashley(P)
3  Belvet(P)
4  Britney(P)
5  Christeen(S)
6  Eve(A)
7  Jane(S)
8  Jennifer(A)
9  Jenny(S)
10 Julia(D)
11 Ketty(A)
12 Kristeen(S)
```