Type of Triangle

Write a query identifying the *type* of each record in the **TRIANGLES** table using its three side lengths. Output one of the following statements for each record in the table:

- **Equilateral**: It's a triangle with *3* sides of equal length.
- **Isosceles**: It's a triangle with *2* sides of equal length.
- **Scalene**: It's a triangle with *3* sides of differing lengths.
- **Not A Triangle**: The given values of *A*, *B*, and *C* don't form a triangle.

Input Format

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

Column	Type
A	Integer
B	Integer
C	Integer

Each row in the table denotes the lengths of each of a triangle's three sides.

Sample Input

A	B	C
20	20	23
20	20	20
20	21	22
13	14	30

Sample Output

Isosceles
Equilateral
Scalene
Not A Triangle

Explanation

Values in the tuple *(20,20,23)* form an Isosceles triangle, because $A = B$.
Values in the tuple *(20,20,20)* form an Equilateral triangle, because $A = B = C$.
Values in the tuple *(20,21,22)* form a Scalene triangle, because $A! = B! = C$.
Values in the tuple *(13,14,30)* cannot form a triangle because the combined value of sides A and B is not larger than that of side C.

SOLUTION –

```
SELECT
CASE
  WHEN ((A + B) > C) AND ((B + C) > A) AND ((C + A) > B) THEN
    CASE
      WHEN A = B AND B = C AND C = A THEN 'Equilateral'
```

```
        WHEN A = B OR B = C OR C = A THEN 'Isosceles'
        ELSE 'Scalene'
      END
   ELSE 'Not A Triangle'
END
FROM TRIANGLES;
```

The PADS

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:
3. There are a total of [occupation_count] [occupation]s.
   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

An **OCCUPATIONS** table that contains the following records:

Name	Occupation
Samantha	Doctor
Julia	Actor
Maria	Actor
Meera	Singer
Ashely	Professor
Ketty	Professor
Christeen	Professor
Jane	Actor
Jenny	Doctor
Priya	Singer

Sample Output

Ashely(P)
Christeen(P)
Jane(A)
Jenny(D)
Julia(A)
Ketty(P)
Maria(A)
Meera(S)
Priya(S)
Samantha(D)
There are a total of 2 doctors.
There are a total of 2 singers.
There are a total of 3 actors.
There are a total of 3 professors.

Explanation

The results of the first query are formatted to the problem description's specifications.
The results of the second query are ascendingly ordered first by number of names corresponding to each profession (*2 <= 2 <= 3 <= 3*), and then alphabetically by profession (*doctor <= singer, and actor <= professor*).

SOLUTION –

```
SELECT CONCAT(NAME,"(",LEFT(OCCUPATION,1),")") FROM OCCUPATIONS ORDER BY NAME;

SELECT CONCAT("There are a total of ", COUNT(OCCUPATION)," ", LOWER(OCCUPATION),"s.")

FROM OCCUPATIONS

GROUP BY OCCUPATION

ORDER BY COUNT(OCCUPATION), OCCUPATION;
```