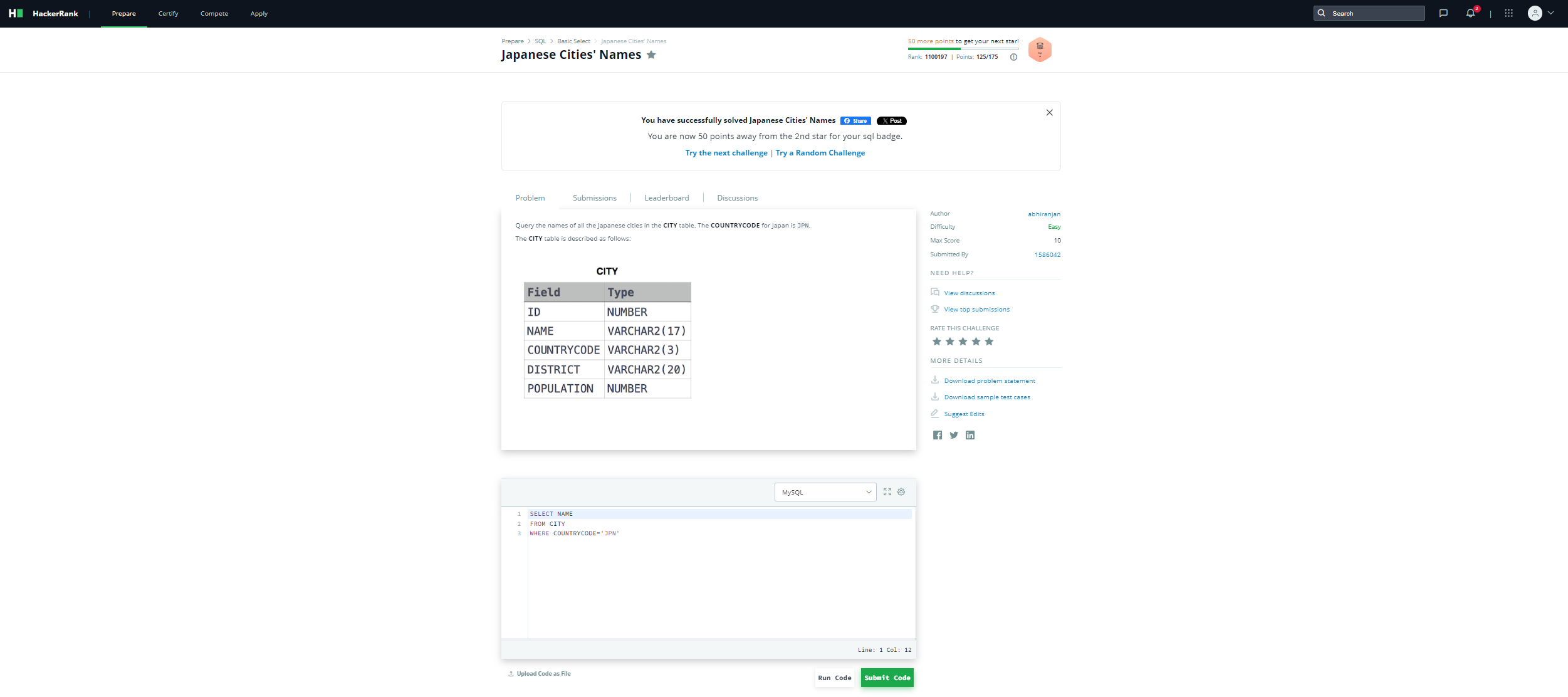
**Question 1 :** [**Japanese Cities' Names**](https://www.hackerrank.com/challenges/japanese-cities-name/problem)

SELECT NAME

FROM CITY

WHERE COUNTRYCODE='JPN'

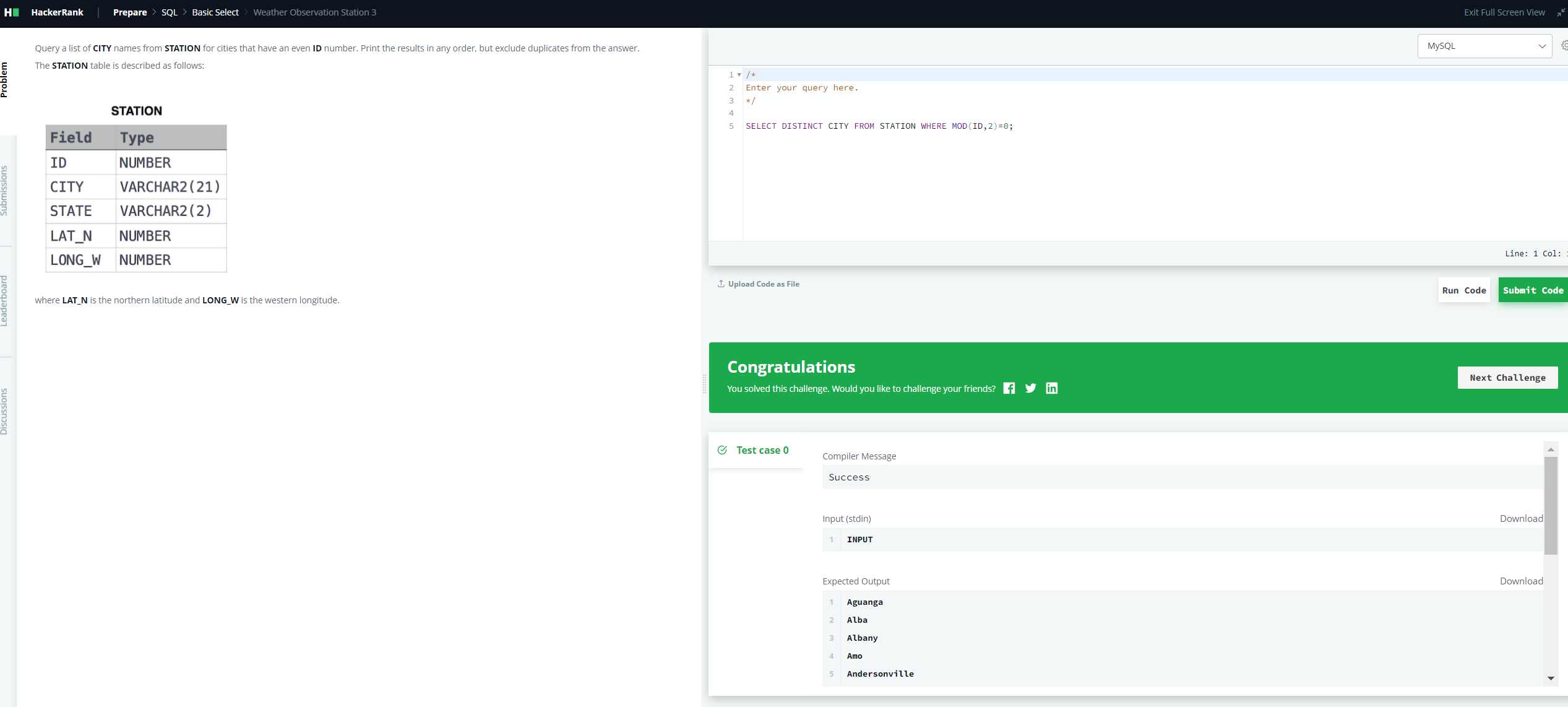


**Question 2:** [**Weather Observation Station 3**](https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true)

SELECT DISTINCT CITY

FROM STATION

WHERE MOD(ID,2) =0;



**Question 3:** [**Weather Observation Station 5**](https://www.hackerrank.com/challenges/weather-observation-station-5/problem)

SELECT

CITY,

LENGTH(CITY) AS Length

FROM

STATION

ORDER BY

LENGTH,

CITY

LIMIT

1;

SELECT

CITY,

LENGTH(CITY) AS Length

FROM

STATION

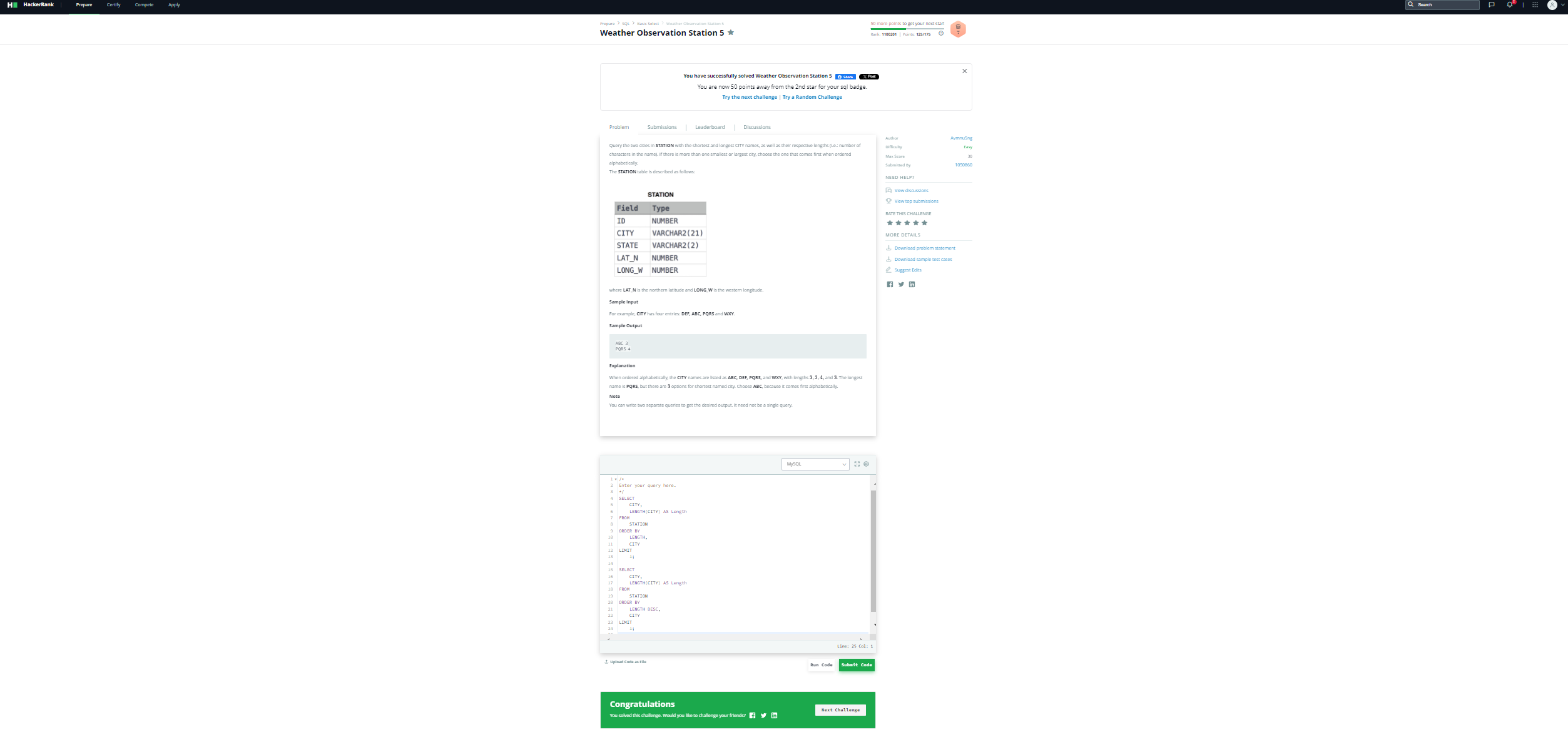
ORDER BY

LENGTH DESC,

CITY

LIMIT

1;

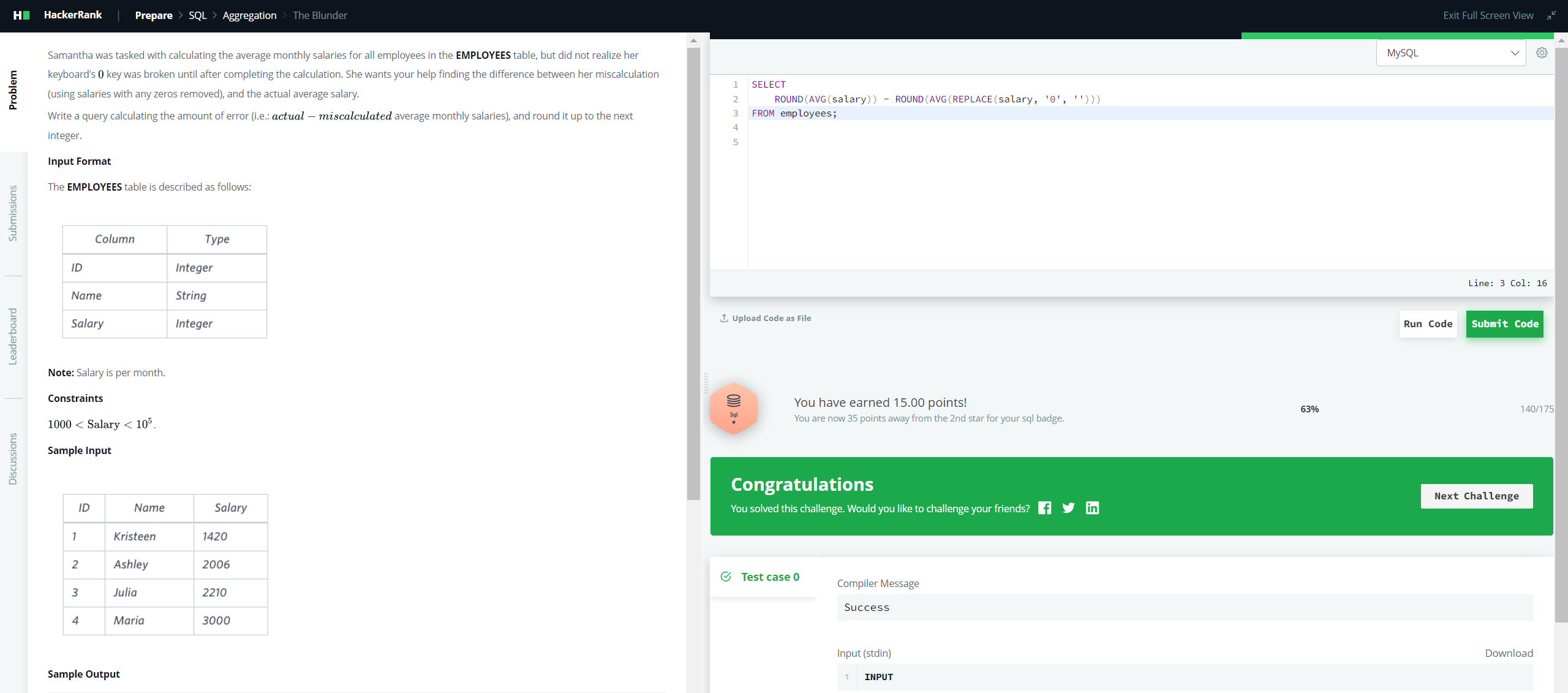


**Question 4:** [**The Blunder**](https://www.hackerrank.com/challenges/the-blunder/problem?isFullScreen=true)

SELECT

ROUND(AVG(salary)) - ROUND(AVG(REPLACE(salary, '0', '')))

FROM employees;

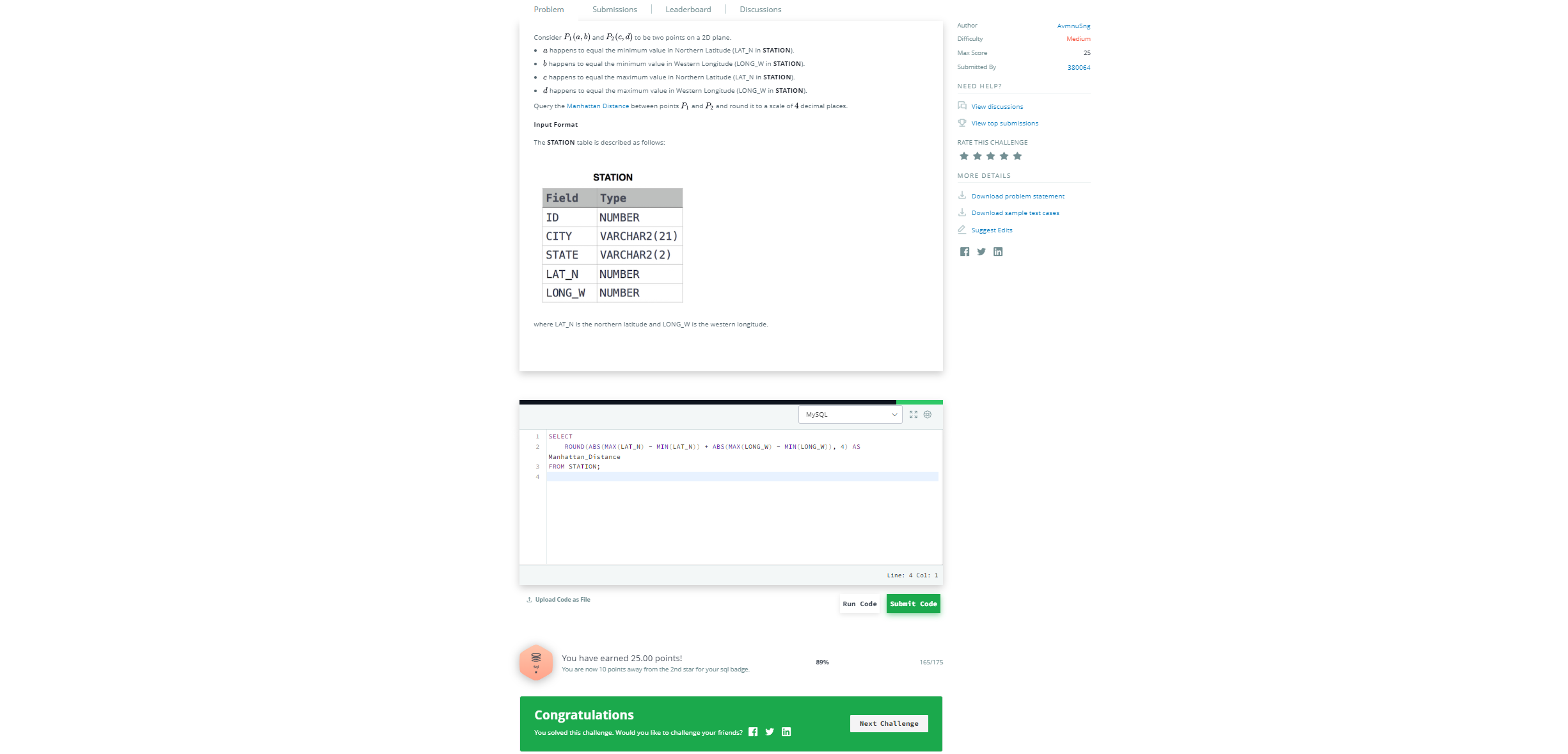


**Question 5:** [**Weather Observation Station 18**](https://www.hackerrank.com/challenges/weather-observation-station-18/problem)

SELECT

ROUND(ABS(MAX(LAT\_N) - MIN(LAT\_N)) + ABS(MAX(LONG\_W) - MIN(LONG\_W)), 4) AS Manhattan\_Distance

FROM STATION;



**Question 6:** [**Average Population of Each Continent**](https://www.hackerrank.com/challenges/average-population-of-each-continent/problem?isFullScreen=true)

SELECT

COUNTRY.Continent,

FLOOR(AVG(CITY.Population)) AS AverageCityPopulation

FROM

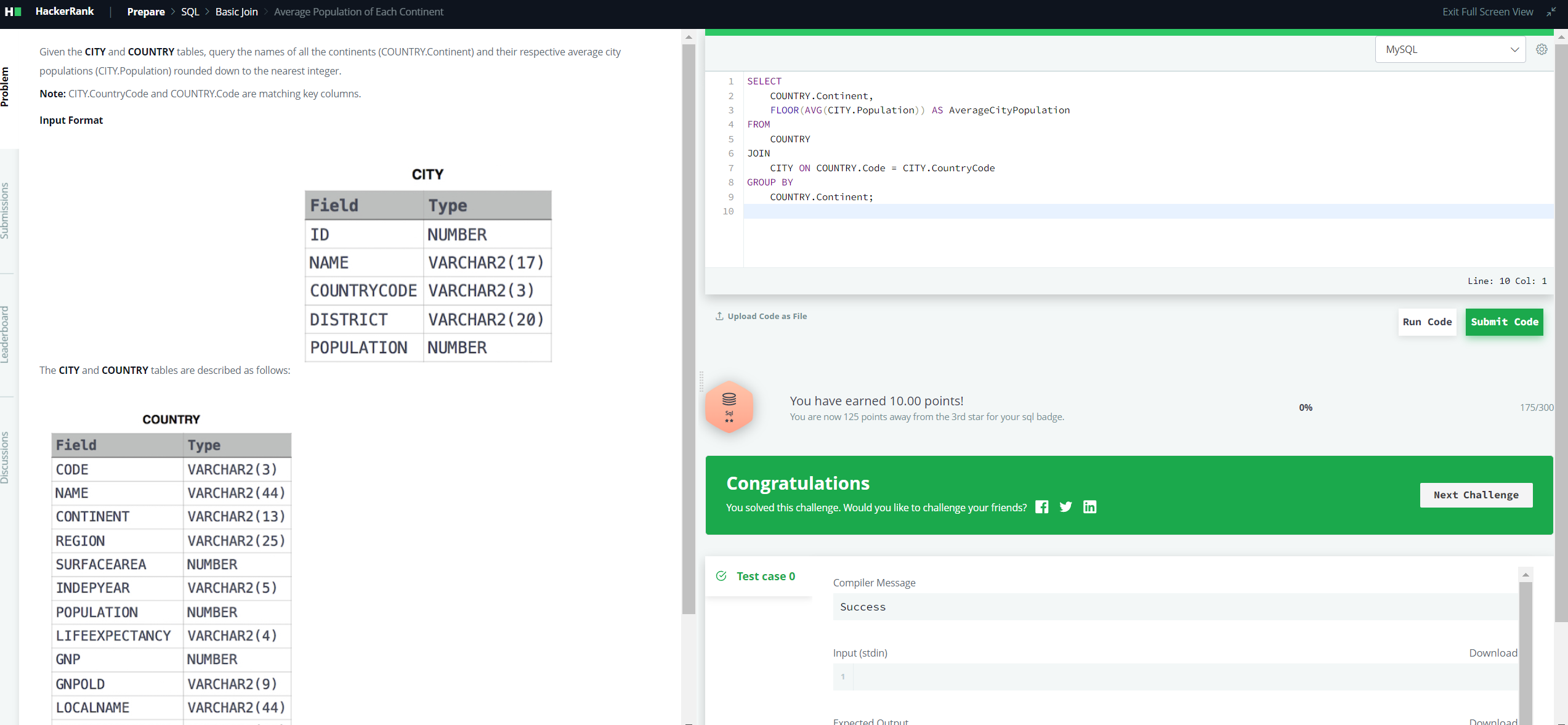
COUNTRY

JOIN

CITY ON COUNTRY.Code = CITY.CountryCode

GROUP BY

COUNTRY.Continent;



**Question 7:** [**The PADS**](https://www.hackerrank.com/challenges/the-pads/problem)

SELECT

CONCAT(Name, '(', LEFT(Occupation, 1), ')')

FROM OCCUPATIONS

ORDER BY Name;

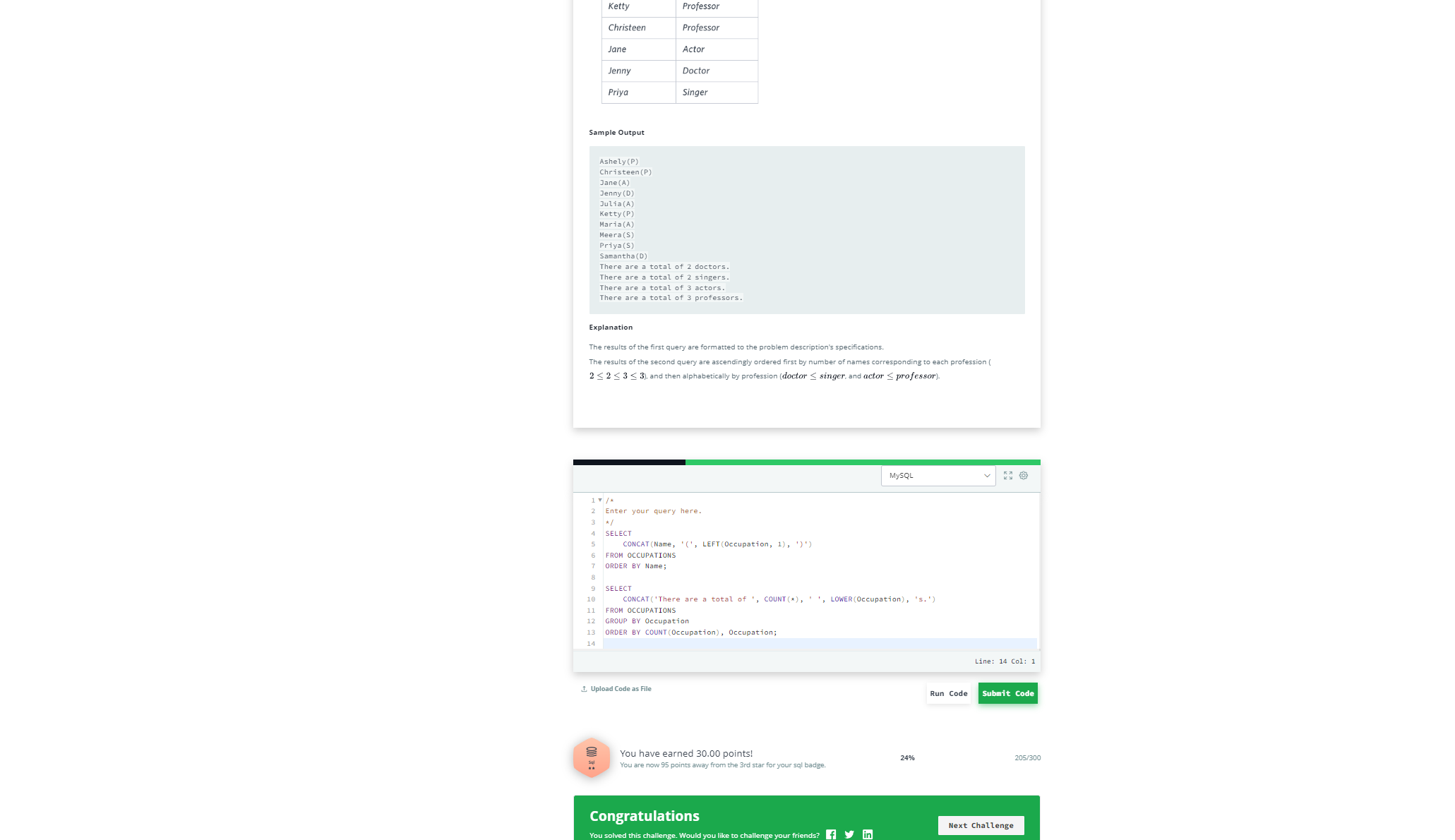
SELECT

CONCAT('There are a total of ', COUNT(\*), ' ', LOWER(Occupation), 's.')

FROM OCCUPATIONS

GROUP BY Occupation

ORDER BY COUNT(Occupation), Occupation;



**Question 8:** [**Type of Triangle**](https://www.hackerrank.com/challenges/what-type-of-triangle/problem) **(Use Case statement)**

SELECT

CASE

WHEN A + B <= C OR A + C <= B OR B + C <= A THEN 'Not A Triangle'

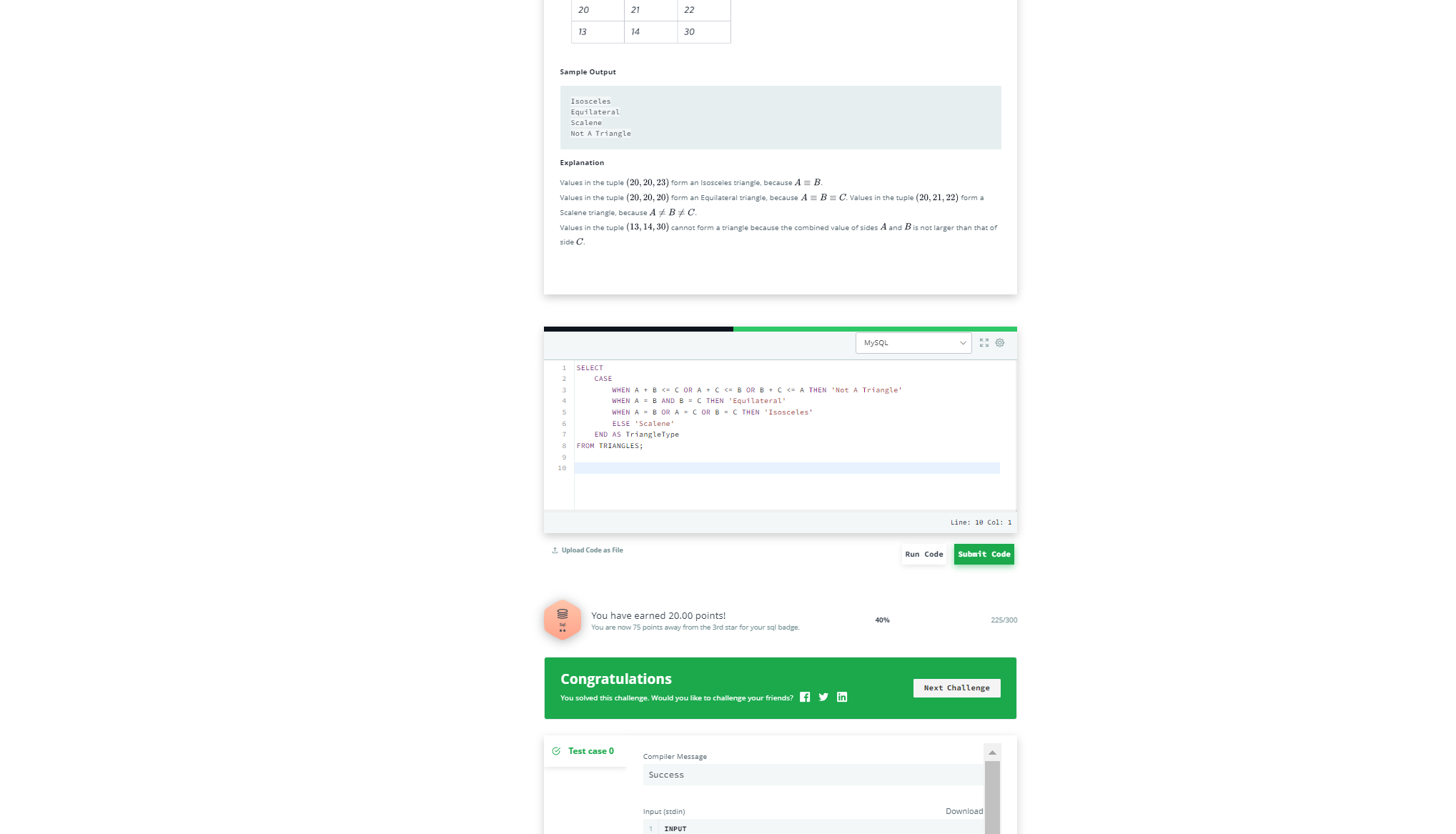
WHEN A = B AND B = C THEN 'Equilateral'

WHEN A = B OR A = C OR B = C THEN 'Isosceles'

ELSE 'Scalene'

END AS TriangleType

FROM TRIANGLES;



**Question 9:** [**Weather Observation Station 13**](https://www.hackerrank.com/challenges/weather-observation-station-13/problem)

SELECT

TRUNCATE(SUM(LAT\_N), 4)

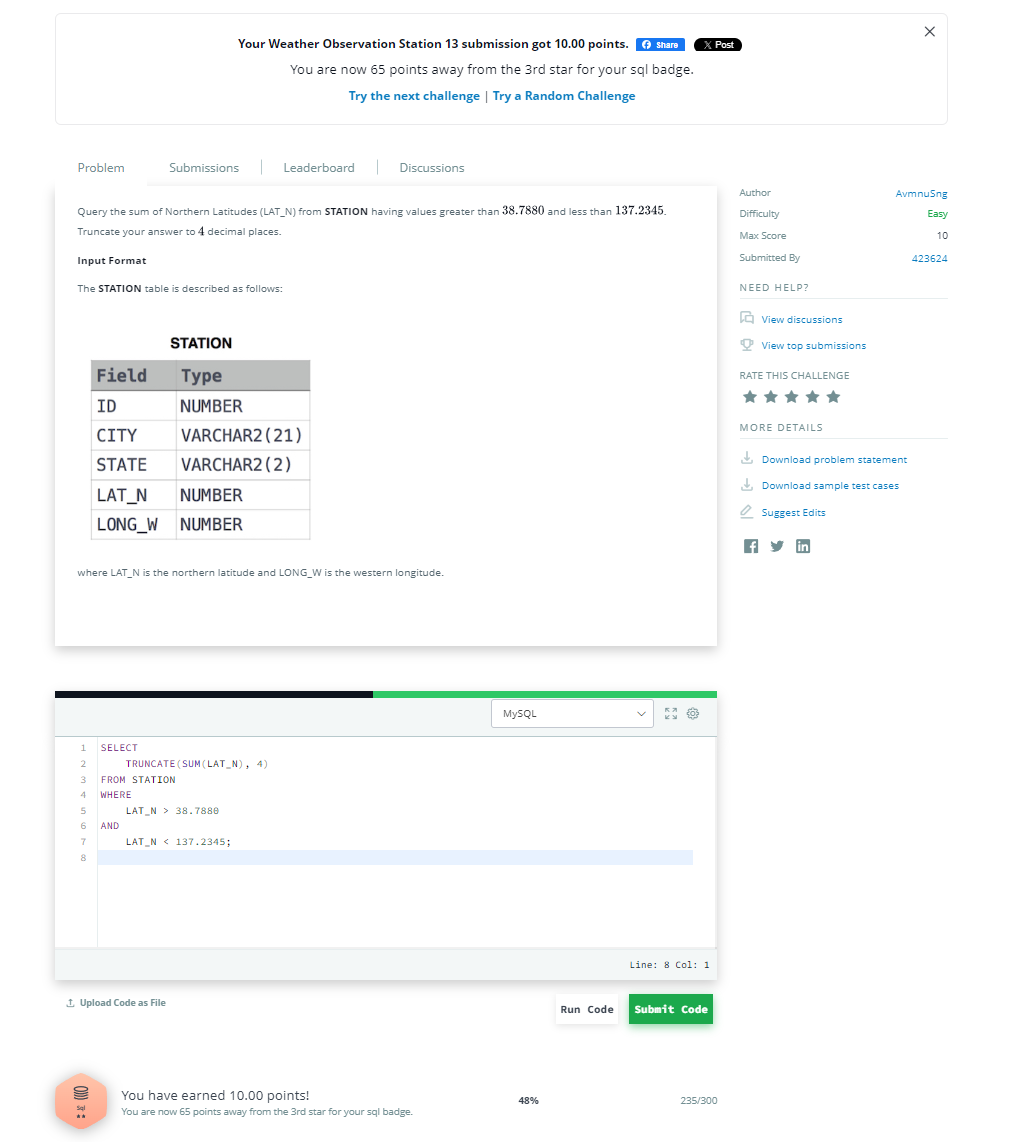
FROM STATION

WHERE

LAT\_N > 38.7880

AND

LAT\_N < 137.2345;



**Question 10:** [**The Report**](https://www.hackerrank.com/challenges/the-report/problem)

SELECT

IF(g.Grade < 8, NULL, s.Name),

g.GRADE,

s.MARKS

FROM

STUDENTS s

INNER JOIN

GRADES g

ON

s.MARKS

BETWEEN

g.MIN\_MARK

AND

g.MAX\_MARK

ORDER BY

g.GRADE DESC,

s.NAME;

