Table: Enrollments

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| student_id    | int     |
| course_id     | int     |
| grade         | int     |
+---------------+---------+
```

(student_id, course_id) is the primary key (combination of columns with unique values) of this table.
grade is never NULL.

Write a solution to find the highest grade with its corresponding course for each student. In case of a tie, you should find the course with the smallest course_id.

Return the result table ordered by student_id in **ascending order**.

The result format is in the following example.

Example 1:

Input:
Enrollments table:
```
+------------+-------------------+
| student_id | course_id | grade |
+------------+-----------+-------+
| 2          | 2         | 95    |
| 2          | 3         | 95    |
| 1          | 1         | 90    |
| 1          | 2         | 99    |
| 3          | 1         | 80    |
| 3          | 2         | 75    |
| 3          | 3         | 82    |
+------------+-----------+-------+
```
Output:
```
+------------+-------------------+
| student_id | course_id | grade |
+------------+-----------+-------+
| 1          | 2         | 99    |
| 2          | 2         | 95    |
| 3          | 3         | 82    |
+------------+-----------+-------+
```

```sql
# Write your MySQL query statement below
WITH CTE AS (SELECT *, DENSE_RANK() OVER (PARTITION BY student_id ORDER BY grade DESC,course_id) R
FROM Enrollments)

SELECT student_id,course_id,grade
FROM CTE
WHERE R=1
ORDER BY student_id
```