Medium

Table: Products

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| product_id    | int     |
| new_price     | int     |
| change_date   | date    |
+---------------+---------+
```

(product_id, change_date) is the primary key (combination of columns with unique values) of this table.
Each row of this table indicates that the price of some product was changed to a new price at some date.

Write a solution to find the prices of all products on 2019-08-16. Assume the price of all products before any change is 10.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

Input:
Products table:

```
+------------+-----------+-------------+
| product_id | new_price | change_date |
+------------+-----------+-------------+
| 1          | 20        | 2019-08-14  |
| 2          | 50        | 2019-08-14  |
| 1          | 30        | 2019-08-15  |
| 1          | 35        | 2019-08-16  |
| 2          | 65        | 2019-08-17  |
| 3          | 20        | 2019-08-18  |
+------------+-----------+-------------+
```

Output:

```
+------------+-------+
| product_id | price |
+------------+-------+
| 2          | 50    |
| 1          | 35    |
| 3          | 10    |
+------------+-------+
```

```
# Write your MySQL query statement below
WITH sub AS (SELECT product_id,
      new_price  AS price,
      change_date,
      ROW_NUMBER() OVER (PARTITION BY product_id ORDER BY change_date DESC) AS rn
FROM Products WHERE change_date<='2019-08-16')

SELECT product_id, price
FROM sub
WHERE rn=1

UNION
```

```
SELECT product_id, 10 AS price
FROM Products
WHERE product_id NOT IN (SELECT product_id
                         FROM sub
                         WHERE rn=1)
```