1164. Product Price at a Given Date

able: 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    |
+------------+-------+
```

To find the prices of all products on a specific date (2019-08-16) given the history of price changes, we need to consider the most recent price change for each product that occurred on or before that date. If a product has no recorded price

change before or on that date, we assume its price to be the default value of 10.

We can do by using SQL query that incorporates a subquery to find the latest price change for each product as of the specified date. Here's how the query can be structured
Write your MySQL query statement below

```
SELECT
    p.product_id,
    COALESCE(
        (SELECT new_price
         FROM Products
         WHERE product_id = p.product_id AND change_date <= '2019-08-16'
         ORDER BY change_date DESC, product_id DESC
         LIMIT 1),
    10) AS price
FROM
    (SELECT DISTINCT product_id FROM Products) AS p;
```