**DESCRIPTION**

Table: Weather

+---------------+---------+

| Column Name | Type |

+---------------+---------+

| id | int |

| recordDate | date |

| temperature | int |

+---------------+---------+

id is the column with unique values for this table.

There are no different rows with the same recordDate.

This table contains information about the temperature on a certain day.

Write a solution to find all dates' id with higher temperatures compared to its previous dates (yesterday).

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

The result format is in the following example.

**Example 1:**

**Input:**

Weather table:

+----+------------+-------------+

| id | recordDate | temperature |

+----+------------+-------------+

| 1 | 2015-01-01 | 10 |

| 2 | 2015-01-02 | 25 |

| 3 | 2015-01-03 | 20 |

| 4 | 2015-01-04 | 30 |

+----+------------+-------------+

**Output:**

+----+

| id |

+----+

| 2 |

| 4 |

+----+

**Explanation:**

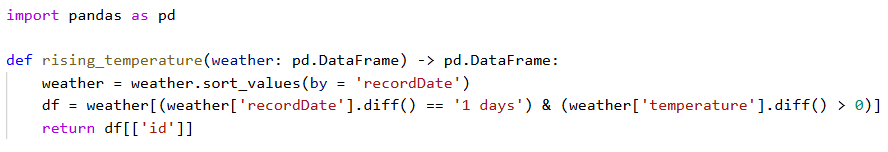
In 2015-01-02, the temperature was higher than the previous day (10 -> 25).

In 2015-01-04, the temperature was higher than the previous day (20 -> 30).

**SOLUTION**

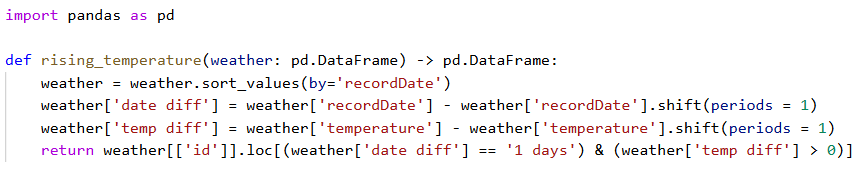
Option 1:

* Sort ‘weather’ by ‘recordDate’ column using **DataFrame.sort\_values**
* Find the following day (‘1 day’ difference) in ‘recordDate’ column and a higher temperature in ‘temperature’ column using **DataFrame.diff**



Option 2:

* Sort ‘weather’ by ‘recordDate’ column using **DataFrame.sort\_values**
* Find two columns ('date diff' and ‘temp diff’) using **DataFrame.shift**



Option 2:

* Sort ‘weather’ by ‘recordDate’ column using **DataFrame.sort\_values**
* Find two columns ('pre\_date' and ‘pre\_temp’) using **DataFrame.shift** with a period of 1
* Return a desired dataframe with ‘id’ column by selecting where date difference is ‘1 days’ and temperature difference is greater than 0

