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 in Pandas

Option 1:

- using **DataFrame.diff**

```
import pandas as pd
def rising_temperature(weather: pd.DataFrame) -> pd.DataFrame:
    weather = weather.sort_values(by = 'recordDate')
    df = weather[(weather['recordDate'].diff() == '1 days') & (weather['temperature'].diff() > 0)]
    return df[['id']]
```

Option 2:

- using **DataFrame.shift**

```
import pandas as pd
def rising_temperature(weather: pd.DataFrame) -> pd.DataFrame:
    weather = weather.sort_values(by='recordDate')
    weather['date diff'] = weather['recordDate'] - weather['recordDate'].shift(periods = 1)
    weather['temp diff'] = weather['temperature'] - weather['temperature'].shift(periods = 1)
    return weather[['id']].loc[(weather['date diff'] == '1 days') & (weather['temp diff'] > 0)]
```