

Assignment 9: Weather Report

กรมอุตุนิยมวิทยาเผยแพร่ข้อมูลผลการตรวจวัดลักษณะอากาศรายวัน ณ สถานีต่าง ๆ ทั่วประเทศซึ่งสามารถ download ได้จากเว็บไซต์ <https://data.tmd.go.th/api/index1.php> หนึ่งในรูปแบบข้อมูลที่เผยแพร่เรียกว่า รูปแบบ json ที่มีโครงสร้างคล้ายกับการเก็บข้อมูลแบบ dict ดังตัวอย่างข้างล่างนี้

```
{
  "Header" : {
    "Title" : "WeatherToday",
    "Description" : "Today's Weather Observation",
    "Uri" : "https://data.tmd.go.th/api/WeatherToday/V1",
    "LastBuiltDate" : "10/3/2021 20:21:27",
    "CopyRight" : "Thai Meteorology Department 2015",
    "Generator" : "TMDDData_API services"
  },
  "Stations" : [
    {
      "WmoNumber" : "48300",
      "StationNameTh" : "แม่ฮ่องสอน",
      "StationNameEng" : "MAE HONG SON",
      "Province" : "แม่ฮ่องสอน",
      "Latitude" : {
        "Value" : "19.2989722222222222222222222222",
        "Unit" : "decimal degree"
      },
      "Longitude" : {
        "Value" : "97.9757777777777777777777777778",
        "Unit" : "decimal degree"
      },
      "Observe" : {
        "Time" : "10/3/2021",
        "MeanSeaLevelPressure" : {
          "Value" : 1015.74,
          "Unit" : "hPa"
        },
        "Temperature" : {
          "Value" : 15.3,
          "Unit" : "celcius"
        },
        "MaxTemperature" : {
          "Value" : 36.7,
          "Unit" : "celcius"
        },
        "DiffMaxTemperature" : {
          "Value" : 0.9,
          "Unit" : "celcius"
        },
        "MinTemperature" : {
          "Value" : 14.8,
          "Unit" : "celcius"
        },
        "DiffMinTemperature" : {
          "Value" : 1.5,
          "Unit" : "hPa"
        },
        "RelativeHumidity" : {
          "Value" : 93,
          "Unit" : "%"
        },
        "WindDirection" : {
          "Value" : "000",
          "Unit" : "degree"
        },
        "WindSpeed" : {
          "Value" : 0,
          "Unit" : "km/h"
        },
        "Rainfall" : {
          "Value" : 0,
          "Unit" : "mm"
        }
      }
    }
  ],
  ...
}
```

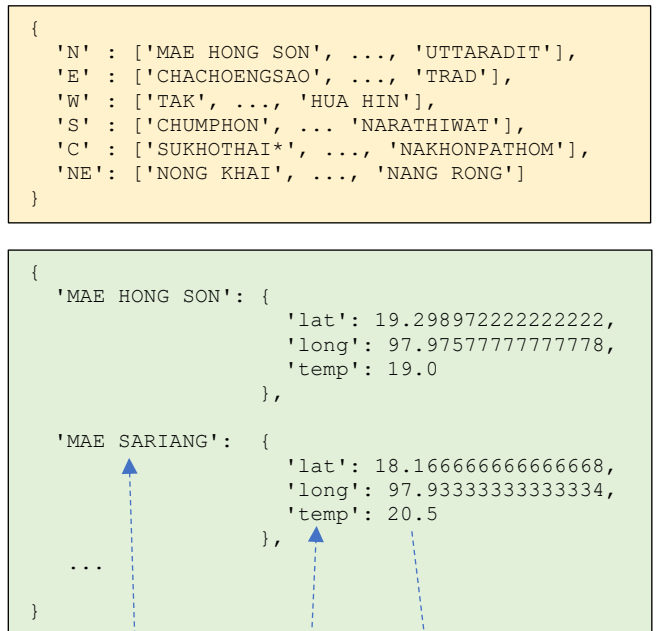
นี่คือรูปแบบของข้อมูลในแฟ้ม weather.json
ไม่ใช่รูปแบบของ dict ที่ใช้ในโจทย์ข้อนี้ ที่จะ
ได้อธิบายในหน้าถัดไป

เพื่อหลีกเลี่ยงการ download และลดความซับซ้อนในการประมวลผลกับข้อมูลข้างบนนี้ ผู้ออกโจทย์ได้ download ข้อมูลมาให้แล้วในแฟ้ม **weather.json** และได้เขียนฟังก์ชัน **read_weather_data** ที่อ่านแฟ้มนี้ (กับอีกแฟ้มชื่อ **provinces.txt** ที่เก็บจังหวัดกับภาคของจังหวัดนั้น) แล้วคืนข้อมูล 3 อย่าง ดังนี้

- วันเวลาที่เก็บข้อมูลสภาพอากาศ เป็นสตริง
- รายชื่อสถานีที่เก็บข้อมูลประจำภาคต่าง ๆ เป็น dict ที่
 - key เป็นชื่อย่อภาค (N, E, W, S, C และ NE)
 - value เป็นลิสต์ที่เก็บชื่อสถานีประจำภาค
- ข้อมูลสภาพอากาศ ณ สถานีต่าง ๆ เป็น dict ที่
 - key เป็นชื่อสถานี
 - value เป็น dict ที่มี 3 คู่ key-value
 - key 'lat' มี value เก็บค่า latitude ของสถานี
 - key 'long' มี value เก็บค่า longitude ของสถานี
 - key 'temp' มี value เก็บค่าอุณหภูมิที่วัดได้ที่สถานี

หมายเหตุ: ชื่อต่าง ๆ ที่เก็บข้างบนเป็นชื่อสถานี ไม่ใช่ชื่อจังหวัด แต่ก็มีชื่อสถานีหลายชื่อที่ตรงกับชื่อจังหวัด

ตัวอย่าง



- ถ้าให้คำสั่ง **date_time, names_in_region, stations = read_weather_data()** ทำงานแล้ว ก็สามารถใช้ข้อมูลใน 3 ตัวแปรทางซ้ายได้ เช่น คำสั่ง **stations['MAE SARIANG']['temp']** ได้ค่า **20.5** ซึ่งคืออุณหภูมิที่สถานี **MAE SARIANG**
- หรือถ้าต้องการอุณหภูมิของสักหนึ่งจังหวัดในภาคอีสาน ก็สามารถใช้คำสั่ง
a_station_name = names_in_region['NE'][0]
t = stations[a_station_name]['temp']

ฟังก์ชันที่มีให้

ผู้ออกโจทย์ได้เขียนฟังก์ชันจำนวนหนึ่งในโปรแกรมต้นฉบับที่ทำงานได้แล้ว ดังนี้

- **read_weather_data()**
คืนวันเวลา, รายชื่อสถานีประจำภาค และข้อมูลสภาพอากาศ ณ สถานีต่าง ๆ ตามที่ได้อธิบายไว้ก่อนหน้านี้ โดยจะอ่านเฉพาะสถานีที่มีข้อมูลสภาพอากาศในแฟ้ม **weather.json** (ถ้าไปดูในแฟ้ม จะพบบางข้อมูลมีค่าเป็น **null** แทนการที่บางสถานี ณ เวลาที่เก็บข้อมูลอาจไม่พร้อมให้ข้อมูล)
- **distance(lat1, long1, lat2, long2)**
คืนระยะทางระหว่างพิกัดละติจูดและลองจิจูด (**lat1, long1**) กับ (**lat2, long2**)
ค่าละติจูดและลองจิจูดเป็นมุมที่มีหน่วยเป็นองศาแบบทศนิยม (decimal degree)
- **show_all_station_names(stations)**
แสดงชื่อสถานีทุกสถานีที่เก็บใน **stations** ทางหน้าจอ
- **plot_map_info(stations, names, date_time)**
แสดงภาพแผนที่ประเทศไทย และแสดงอุณหภูมิจากข้อมูลสภาพอากาศใน **stations** เฉพาะของสถานีที่มีชื่อใน **names** โดยแสดงวันเวลาที่เก็บใน **date_time** บนหัวเรื่อง
- **main()**
เป็นฟังก์ชันหลักแสดงรายการทางเลือกให้โปรแกรมทำงาน มี 5 ทางเลือกดังตัวอย่างในหน้าถัดไป

ตัวอย่างการสั่งทำงาน

```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

Select 0,1,2,3,4,5: 0

Amnat Charoen, Aranya Prathet, Ayutthaya, Bang Na Agromet.
 Bangkok Port (Klong Toei), Bhumibol Dam, Bua Chum, Bueng Kan
 Chachoengsao, Chainat Agromet., Chaing Rai Agromet., Chaiyaphum
 Chawang, Chiang Mai, Chiang Rai, Chok Chai
 Chumphon, Doi Ang Kang, Doi Mu Soe Agromet., Don Muang Airport
 Hua Hin, Huai Pong Agromet., Kabin Buri, Kalasin
 Kanchana Buri, Kho Hong Agromet., Khon Kaen, Ko Lanta
 Ko Sichang, Krabi, Laem Chabang, Lampang
 Lamphun, Loei, Loei Agromet., Lom Sak
 Mae Hong Son, Mae Sariang, Mae Sot, Mahasarakham
 Nakhon Phanom, Nakhon Phanom Agromet., Nakhon Ratchasima, Nakhon Sawan
 Nakhonsi Thammarat, Nakhonsi Thammarat Agromet., Nan, Nan Agromet.
 Narathiwat, Nong Bua Lam Phu, Nong Khai, Nong Phlub Agromet.
 Pathumthani, Pattani Airport, Phatthalung Agromet., Phatthaya
 Phetcha Buri, Phetchabun, Phitsanulok, Phliu Agromet.
 Phrae, Phuket, Phuket Airport, Pichit Agromet.
 Prachin Buri, Prachuap Khirikhan, Ranong, Ratcha Buri
 Roi Et, Roi Et Agromet., Sa Dao, Sa Kaew
 Sakon Nakhon Agromet., Samut Prakarn, Samutsongkram, Sattahip
 Sawi Agromet., Si Saket Agromet., Si Samrong Agromet., Songkhla
 Suphan Buri, Surat Thani, Surat Thani Agromet., Surin
 Suvarnabhumi Airport, Tak, Takfa Agromet., Takua Pa
 Tha Tum, Tha Wangpha, Thoen, Thong Phaphum
 Trad, Trang Airport, U Thong Agromet., Ubon Ratchathani
 Udon Thani, Umphang, Uthaithani, Uttaradit
 Yala Agromet.

ตัวสีแดงคือข้อมูลที่ผู้ใช้ป้อนให้เป็น input
 ตัวพื้นเหลืองและภาพแผนที่คือผลลัพธ์ที่ได้

ทางเลือก 0 แสดงชื่อสถานีทั้งหมด
 ไม่ได้แสดงภาพแผนที่

```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

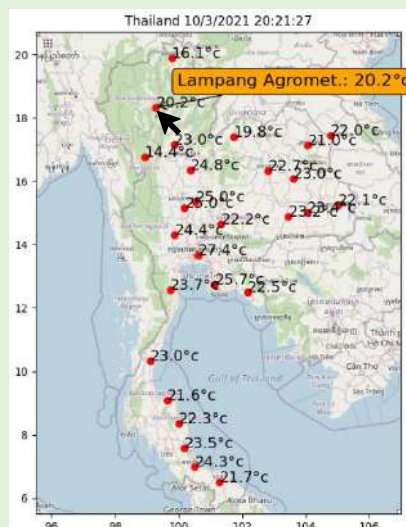
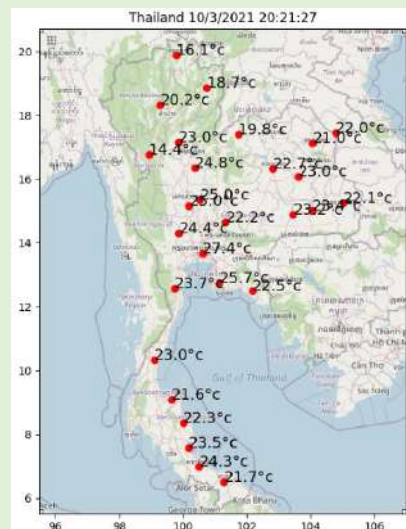
Select 0,1,2,3,4,5: 1

Station name: agro

Chaing Rai Agromet. 16.1°C
 Lampang Agromet. 20.2°C
 Nan Agromet. 18.7°C
 Loei Agromet. 19.8°C
 Sakon Nakhon Agromet. 21.0°C
 Nakhon Phanom Agromet. 22.0°C
 Si Samrong Agromet. 23.0°C
 Doi Mu Soe Agromet. 14.4°C
 Tha Phra Agromet. 22.7°C
 Pichit Agromet. 24.8°C
 Takfa Agromet. 25.0°C
 Chainat Agromet. 25.0°C
 Roi Et Agromet. 23.0°C
 Ubon Ratchathani Agromet. 22.1°C
 Si Saket Agromet. 23.4°C
 U Thong Agromet. 24.4°C
 Pakchong Agromet. 22.2°C
 Surin Agromet. 23.2°C
 Bang Na Agromet. 27.4°C
 Huai Pong Agromet. 25.7°C
 Phliu Agromet. 22.5°C
 Nong Phlub Agromet. 23.7°C
 Sawi Agromet. 23.0°C
 Surat Thani Agromet. 21.6°C
 Nakhonsi Thammarat Agromet. 22.3°C
 Phatthalung Agromet. 23.5°C
 Kho Hong Agromet. 24.3°C
 Yala Agromet. 21.7°C

ทางเลือกที่ 1 แสดงสถานีที่มีสตริง
 ที่ให้เป็นส่วนหนึ่งของชื่อสถานี
 เช่น ในตัวอย่าง ได้ agro

ในกรณีที่อยากรู้ชื่อสถานีของจุด
 ในแผนที่ให้นำเมาส์ไปวางไว้ที่
 จุดที่ต้องการให้แสดงชื่อ



```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

Select 0,1,2,3,4,5: **2**

K: **2**

Doi Mu Soe Agromet. 14.4°C

Mae Hong Son 15.3°C

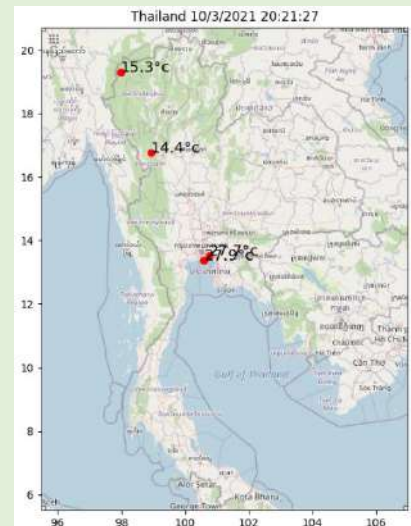
Pilot Station 27.9°C

Samut Prakarn 27.7°C

ทางเลือกที่ 2

แสดงสถานีที่มีอุณหภูมิสูงสุด K สถานี

และสถานีที่มีอุณหภูมิต่ำสุด K สถานี



```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

Select 0,1,2,3,4,5: **3**

Narathiwat 23.4°C

Chiang Rai 17.5°C

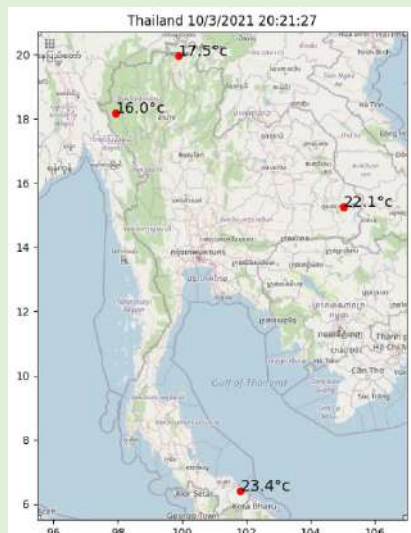
Mae Sariang 16.0°C

Ubon Ratchathani Agromet. 22.1°C

ทางเลือกที่ 3

แสดงสถานีที่อยู่ใกล้จุดน้อยสุด และมากที่สุด

และสถานีที่อยู่ใกล้จุดน้อยสุด และมากที่สุด



```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

Select 0,1,2,3,4,5: **4**

Main station: **Mae hong son**

How many nearby stations? **3**

Mae Hong Son 15.3°C

Chiang Mai 18.3°C

Mae Sariang 16.0°C

Doi Ang Kang 18.9°C

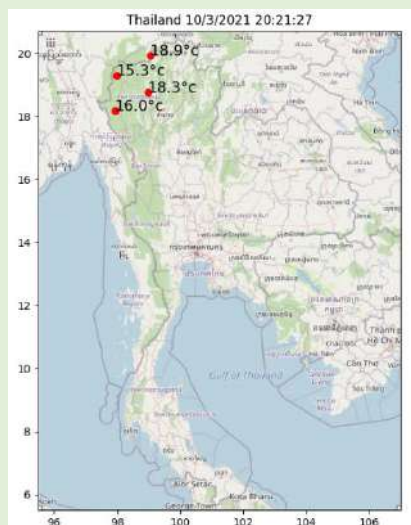
ทางเลือกที่ 4

แสดงสถานีที่อยู่ใกล้สถานี

Mae hong son ที่สุด 3 สถานี

รวมของ Mae hong son ด้วย

เป็น 4 สถานี



```
>>> %Run prog09_weather.py
```

Weather Report:

- (0) List all stations
- (1) Temperatures at selected stations
- (2) Top K max & min temperature stations
- (3) Temperatures at the peak stations
- (4) Temperatures at the nearby stations
- (5) Average temperatures in the region

Select 0,1,2,3,4,5: **5**

Region (N,E,W,S,C,NE): **W**

Average temperature = **22.1°C**

Tak **21.1°C**

Mae Sot **19.5°C**

Bhumibol Dam **21.0°C**

Doi Mu Soe Agromet. **14.4°C**

Umphang **16.0°C**

Ratcha Buri **24.0°C**

Kanchana Buri **25.3°C**

Thong Phaphum **23.8°C**

Phetcha Buri **25.5°C**

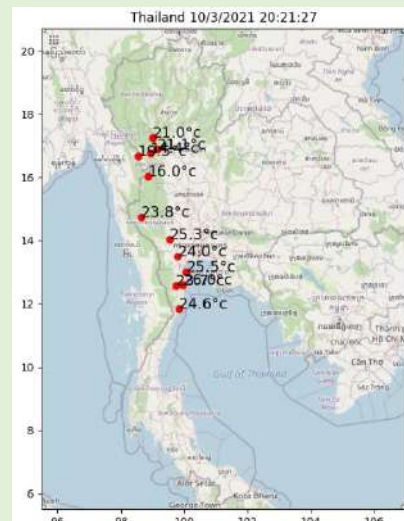
Prachuap Khirikhan **24.6°C**

Hua Hin **26.0°C**

Nong Phlub Agromet. **23.7°C**

ทางเลือกที่ 5

แสดงทุกสถานีในภาคใต้ (S) และแสดงอุณหภูมิเฉลี่ยของทุกสถานีในภาคใต้



ฟังก์ชันที่ต้องเขียน (stations ที่ส่งให้ทุกฟังก์ชัน มีอย่างน้อย 1 สถานีแน่ ๆ)

```
def approx_match(stations, name)
```

- # stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อน
- # name เป็นสตริง
- # คืนลิสต์ที่เก็บชื่อสถานีที่มีตัวอักษรใน name เป็นส่วนหนึ่งของชื่อสถานี โดยไม่สนใจว่า
- # เป็นตัวพิมพ์เล็กหรือใหญ่ และไม่สนใจเว้นวรรคภายในด้วย (ชื่อในลิสต์ที่คืนเป็นผลลัพธ์เรียงอย่างไรก็ได้)
- #หมายเหตุ: ชื่อสถานีที่เป็นคีย์ใน stations นั้นเป็นตัวพิมพ์ใหญ่หมด ผลที่คืนจากฟังก์ชันนี้ก็จะเก็บตัวพิมพ์ใหญ่
- # แต่ตอนค้น name ใน stations นั้น ต้องเป็นแบบไม่สนใจว่าเป็นตัวพิมพ์เล็กหรือใหญ่
- # เช่น name = 'Tha ni' จะได้ผลลัพธ์เป็นดังนี้ (เรียงลำดับอื่นก็ได้)
- # ['UDON THANI', 'UBON RATCHATHANI AGROMET.', 'UBON RATCHATHANI',
- # 'UTHAITHANI', 'PATHUMTHANI', 'SURAT THANI', 'SURAT THANI AGROMET.']

```
def top_k_min_temp_stations(stations, K)
```

- # stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อน, K เป็นจำนวนเต็มบวก
- # คืนลิสต์ของชื่อสถานีที่มีอุณหภูมิต่ำสุด K สถานี ให้เรียงจากซ้ายไปขวาตามสถานีที่มีอุณหภูมิต่ำสุดไปสูงขึ้น
- # ในกรณีที่อุณหภูมิเท่ากัน ให้เลือกสถานีที่มีชื่อมาก่อนตามลำดับในพจนานุกรม ในกรณีที่ไม่มีถึง K สถานี ก็คืนเท่าที่มี

```
def top_k_max_temp_stations(stations, K)
```

- # stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อน, K เป็นจำนวนเต็มบวก
- # คืนลิสต์ของชื่อสถานีที่มีอุณหภูมิสูงสุด K สถานี ให้เรียงจากซ้ายไปขวาตามสถานีที่มีอุณหภูมิสูงสุดไปต่ำลง
- # ในกรณีที่อุณหภูมิเท่ากัน ให้เลือกสถานีที่มีชื่อมาก่อนตามลำดับในพจนานุกรม ในกรณีที่ไม่มีถึง K สถานี ก็คืนเท่าที่มี

```
def peak_stations(stations)
```

- # stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อน
- # คืน ลิสต์ของชื่อสถานีที่มี ละติจูดน้อยสุด ละติจูดมากที่สุด ลองจิจูดน้อยสุด และลองจิจูดมากที่สุด (ต้องคืน 4 ชื่อถึงแม้จะซ้ำ)


```
def k_nearby_stations(stations, main_station, K)
```

```
# stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อนนี้
```

```
# main_station เก็บชื่อสถานีที่มีใน stations และ K เป็นจำนวนเต็มบวก
```

```
# คืนลิสต์ของชื่อสถานีที่อยู่ใกล้กับสถานี main_station ที่สุด เป็นจำนวน K สถานี
```

```
# ให้เรียงจากซ้ายไปขวาตามสถานีที่อยู่ใกล้สุดไปไกลขึ้น
```

```
# ในกรณีที่ระยะเท่ากัน ให้เลือกสถานีที่มีชื่อมาก่อนตามลำดับในพจนานุกรม ในกรณีที่ไม่มีถึง K สถานี ก็คืนเท่าที่มี
```

```
def average_temp(stations, names)
```

```
# stations เป็น dict เก็บข้อมูลพิกัดและอุณหภูมิของสถานีต่าง ๆ ในรูปแบบที่อธิบายมาก่อนนี้
```

```
# names เป็นลิสต์ที่เก็บชื่อสถานีต่าง ๆ ที่มีใน stations โดยที่ names มีอย่างน้อยหนึ่งชื่อ
```

```
# คืน อุณหภูมิเฉลี่ยของสถานีต่าง ๆ ที่มีชื่อเก็บใน names
```

โปรแกรมต้นฉบับ

```
# Prog-09: Weather Report
```

```
# 6?3????21 Name ?
```

ใส่เลขประจำตัว ชื่อ นามสกุล

```
import math
import matplotlib.pyplot as plt
import json

def read_weather_data():
    ...
    return date_time, names_in_region, stations

def distance(lat1, long1, lat2, long2):
    ...
    return d

def plot_map_info(stations, names, date_time):
    ...
    plt.show()

def show_all_station_names(stations):
    ...

def main():
    while True:
        print("Weather Report:\n" +
              " (0) List all stations\n" +
              " (1) Temperatures at selected stations\n" +
              " (2) Top K max & min temperature stations\n" +
              " (3) Temperatures at the peak stations\n" +
              " (4) Temperatures at the nearby stations\n" +
              " (5) Average temperatures in the region")
        choice = input('Select 0,1,2,3,4,5: ')
        if '0' <= choice <= '5': break
        print('Try again.')

    date_time, names_in_region, stations = \
        read_weather_data()
    names = []

    if choice == '0':
        ...
    elif choice == '1':
        ...
    elif choice == '2':
        ...
    elif choice == '3':
        ...
    elif choice == '4':
        ...
    elif choice == '5':
        ...
    if len(names) > 0:
        plot_map_info(stations, names, date_time)
# -----
```

```
def approx_match(stations, name):
```

```
def top_k_min_temp_stations(stations, K):
```

```
def top_k_max_temp_stations(stations, K):
```

```
def peak_stations(stations):
```

```
def k_nearby_stations(stations, main_station, K):
```

```
def average_temp(stations, names):
```

```
# -----
main()
```

โปรแกรมที่ส่ง ห้ามเปลี่ยนอะไรใด ๆ ในส่วนที่มีพื้น
หลังเป็นสีแดง และที่เป็นตัวสีแดงโดยเด็ดขาด
เปลี่ยนได้เฉพาะส่วนที่มีพื้นหลังเป็นสีเขียวเท่านั้น

ข้อเสนอแนะ

- การตรวจให้คะแนนจะใช้วิธีเรียกแต่ละฟังก์ชันที่เขียน โดยส่งพารามิเตอร์ต่าง ๆ ให้กับฟังก์ชัน ค่าในพารามิเตอร์อาจไม่เหมือนข้อมูลที่อ่านและแปลงจากแฟ้ม weather.json ก็ได้ (แต่มีรูปแบบการจัดเก็บเหมือนที่ได้อธิบายไว้) เช่น

```
stations = { 'A Bc': {'lat': 10.20, 'long': 2.51, 'temp': 14.3},
             'CbCD': {'lat': 10.21, 'long': 2.49, 'temp': 14.5},
             'AZ':   {'lat': 10.22, 'long': 2.23, 'temp': 15.1} }
name = 'b c'
names = approx_match( stations, name )
```

ก็ควรได้ names มีค่าเป็น ['A Bc', 'CbCD'] (หรือ ['CbCD', 'A Bc'] ก็ได้)

- การทำงานในแต่ละฟังก์ชัน ไม่มีการ print อะไรทั้งสิ้น ข้อมูลและภาพแผนที่ที่แสดงเกิดจากฟังก์ชัน main และ plot_map_info
- เมื่อใดที่ต้องคำนวณระยะทางระหว่างสองสถานีใด ๆ ให้เรียกใช้ฟังก์ชัน distance ที่เขียนให้แล้ว ห้ามเขียนวิธีคำนวณระยะทางเอง
- แต่ละฟังก์ชันที่เขียน เป็นฟังก์ชันที่ถูกเรียกใช้ในแต่ละทางเลือก เมื่อสั่งโปรแกรมทำงาน ดังตารางข้างล่างนี้

ทางเลือก	ฟังก์ชันที่เรียกใช้
1	approx_match
2	top_k_min_temp_stations top_k_max_temp_stations
3	peak_stations
4	k_nearby_stations
5	average_temp

หมายเหตุ: โปรแกรมต้นฉบับที่ให้ไปมีคำสั่งที่ทำงานให้กับทางเลือกที่ 0 ไว้เรียบร้อยแล้ว

ดังนั้นเมื่อเขียนฟังก์ชันในคอลัมน์ขวาของแถวใดเสร็จ ก็สามารถสั่งโปรแกรมทำงานแล้วเลือกทางเลือกทางซ้าย เพื่อทดลองดูว่าทำงานได้ตรงตามที่คาดหรือไม่

- ควรทดสอบทีละฟังก์ชันโดยสร้างข้อมูลให้กับพารามิเตอร์ต่าง ๆ ตามข้อกำหนดของฟังก์ชันด้วย