

## ▼ 2110101: การบ้านครั้งที่ 6: เรื่องลิสต์ (List Processing)

### โปรแกรมบันทึกการบริโภคอาหาร

เราจะเขียนโปรแกรมบันทึกการบริโภคอาหาร ที่จะคำนวณคุณค่าและปริมาณของสารอาหาร ที่ได้รับจากการบริโภคในแต่ละวัน โดยเรามีตารางข้อมูลโภชนาการ (Nutrients) ประกอบไปด้วยข้อมูล รหัสอาหาร ชื่ออาหาร ค่าพลังงาน(kcal) ปริมาณ คาร์โบไฮเดรต(g) โปรตีน(g) ไขมัน(g) น้ำตาล(g) โซเดียม(mg) ตามลำดับ ซึ่งเก็บอยู่ในลิสต์ของลิสต์ในรูปแบบเป็น

[food\_id, food\_name, kcal, carb, protein, fat, sugar, sodium] โดยมีตัวอย่างของลิสต์เป็นดังนี้

```
[['R010007', 'ลาบ หมู', 267, 17, 23, 12, 2, 1470],
['R010014', 'ส้มตำ ไทย', 143, 143, 31, 5, 27, 1064],
['R010005', 'ก๋วยเตี๋ยว ผัดไทย ใส่ ไข่', 447, 49, 21, 18, 'NA', 1139],
['P010019', 'ขนมจีน ชวน้ำ', 437, 62, 9, 17, 'NA', 810],
['P010021', 'ขนมจีน น้ำยา', 348, 41, 14, 14, 11, 1210],
['P010023', 'ขนมจีน น้ำพริก', 497, 75, 11, 17, 'NA', 'NA'],
['P010041', 'ข้าว หมก ไก่', 481, 74, 19, 12, 'NA', 900],
['P020008', 'ข้าว ราด กะเพรา ไก่', 458, 60, 20, 15, 'NA', 1200],
['P010049', 'ข้าว ไข่ พะโล้', 464, 55, 20, 18, 'NA', 946],
['P010025', 'ข้าว ไก่ ผัด กะเพรา', 432, 54, 20, 15, 'NA', 'NA'],
['F010003', 'กล้วย ไข่', 62, 14, 0, 0, 7, 4],
['D010032', 'บัวลอย เผือก', 336, 62, 2, 9, 'NA', 'NA'],
['D010033', 'ลาบ เผือก', 'NA', 'NA', 'NA', 'NA', 'NA', 'NA']]
```

### หมายเหตุ

- ชื่ออาหารเป็น string ที่ตัดคำมาให้แล้ว โดยใช้ วรรค คั่นระหว่างคำ เช่น 'ข้าวหมกไก่' ประกอบด้วย 3 คำ คือ 'ข้าว' 'หมก' 'ไก่' จึงได้ string ชื่ออาหารเป็น 'ข้าว หมก ไก่' เป็นต้น
- ถ้าไม่มีข้อมูลปริมาณสารอาหารจะใส่ 'NA'

### งานที่ 1 หาค่าความคล้ายระหว่างชื่ออาหาร

ชื่ออาหารที่ผู้บริโภคบันทึกว่ารับประทาน อาจจะไม่เหมือนชื่ออาหารในตารางโภชนาการ เป๊ะ ๆ เช่น ผู้บริโภคบันทึกว่ารับประทาน 'ข้าว กะเพรา ไก่' ดังนั้น เราต้องคำนวณหาความคล้ายของชื่ออาหาร โดยให้

ค่าความคล้าย = จำนวนคำที่ตรงกัน\*2 / ผลรวมจำนวนคำของชื่ออาหารทั้งสอง

เช่น

- ค่าความคล้ายของชื่ออาหาร 'ข้าว กะเพรา ไก่' กับ 'ข้าว ไก่ ผัด กะเพรา' =  $3*2/7 \sim 0.857$
- ค่าความคล้ายของชื่ออาหาร 'ข้าว กะเพรา ไก่' กับ 'ข้าว หมก ไก่' =  $2*2/6 \sim 0.667$

ให้เขียนฟังก์ชัน compute\_similarity(food\_name1, food\_name2) ที่

- รับ food\_name1 และ food\_name2 เป็นสตริงเก็บชื่ออาหารที่ตัดคำมาให้แล้ว โดยใช้ วรรค คั่นระหว่างคำ
- คืน ค่าความคล้ายของชื่ออาหาร ซึ่งเป็นเลขจำนวนจริงที่มีค่าระหว่าง [0,1]

ตัวอย่างเช่น หากเรียกใช้ฟังก์ชันนี้โดยคำสั่ง

```
print(compute_similarity('ข้าว กะเพรา ไก่', 'ข้าว ไก่ ผัด กะเพรา'))
print(compute_similarity('กล้วย ไข่', 'ข้าว ไก่ ผัด กะเพรา'))
print(compute_similarity('ข้าว ไข่ พะโล้', 'ข้าว ไก่ ผัด กะเพรา'))
print(compute_similarity('ข้าว ผัด กะเพรา ไก่', 'ข้าว ไก่ ผัด กะเพรา'))
```

จะได้ผลลัพธ์เป็น

```
0.8571428571428571
0.0
0.2857142857142857
1.0
```

## งานที่ 2 หาทห้สอาหาร

ให้เขียนฟังก์ชัน match\_foods(nutrient, food\_name) ที่

- รับ nutrient เก็บตารางข้อมูลโภชนาการรูปแบบของลิสต์ที่อธิบายข้างบน
- รับ food\_name เป็นสตริงชื่ออาหาร
- คืน ลิสต์ผลลัพธ์ที่เก็บรหัสอาหาร food\_id ที่มีชื่ออาหารตรงกับ food\_name มากที่สุด โดยพิจารณา ดังนี้
  - มีค่าความคล้ายของชื่ออาหารคำนวณจาก compute\_similarity() สูงสุด และ ค่าความคล้ายต้องมากกว่า 0.5 มิเช่นนั้นคืนลิสต์ว่าง
  - ถ้าค่าความคล้ายสูงสุดมีค่าเท่ากันมากกว่า 1 รายการ และมีค่ามากกว่า 0.5 ให้คืนลิสต์ของ food\_id ทุกรายการที่มีค่าความคล้ายสูงสุดซึ่งมีค่าเท่ากัน และเรียงรายการที่มีความคล้ายเท่ากันตาม food\_id เรียงลำดับจากน้อยไปมากตามพจนานุกรม

ตัวอย่างเช่น หากเรียกใช้ฟังก์ชันนี้โดยคำสั่ง

```
matched_foods = match_foods(nutrient, 'ข้าว กะเพรา ไก่')
```

matched\_foods จะมีค่าเป็น

```
['P010025', 'P020008']
```

เพราะ compute\_similarity('ข้าว กะเพรา ไก่', 'ข้าว ไก่ ผัด กะเพรา') กับ compute\_similarity('ข้าว กะเพรา

ไก่', 'ข้าว รัต กะเพรา ไก่') ได้ค่าเท่ากัน และเป็นค่ามากที่สุด

```
matched_foods = match_foods(nutrient, 'ห่อ หมก')
```

matched\_foods จะมีค่าเป็น

```
[]
```

เพราะ compute\_similarity('ห่อ หมก', 'ข้าว หมก ไก่') มีค่ามากที่สุดก็จริง แต่น้อยกว่า 0.5

### งานที่ 3 คำนวณค่าพลังงานและสารอาหารจากชื่ออาหาร

ให้เขียนฟังก์ชัน get\_nutrient(nutrient, food\_name) ที่

- รับ nutrient เก็บตารางข้อมูลโภชนาการรูปแบบของลิสต์
- รับ food\_name เป็นสตริงชื่ออาหาร
- คืน ลิสต์ผลลัพธ์ที่เก็บข้อมูลพลังงานและสารอาหาร ในรูปแบบ [kcal, carb, protein, fat, sugar, sodium] ซึ่งคำนวณจากผลลัพธ์จากการเรียกฟังก์ชัน match\_foods()
  - ในกรณีที่ food\_name ไม่คล้ายกับชื่ออาหารในตารางโภชนาการ nutrient (match\_foods() คืนค่าลิสต์ว่าง) ให้คืนค่าลิสต์ว่าง
  - ในกรณีที่ food\_name คล้ายกับชื่ออาหารในตารางโภชนาการ nutrient มากกว่า 1 รายการ ให้ใช้ ค่าเฉลี่ย ของปริมาณพลังงานและสารอาหารต่าง ๆ
  - ปริมาณพลังงานและสารอาหารที่มีค่าเป็น NA ให้คงค่า NA และไม่นำไปคิดรวมในการคำนวณค่าเฉลี่ย
  - ปริมาณพลังงานและสารอาหารที่ไม่ใช่ 'NA' ให้เป็นเลขจำนวนจริง

ตัวอย่างเช่น หากเรียกใช้ฟังก์ชันนี้โดยคำสั่ง

```
print(get_nutrient(nutrient, 'ห่อ หมก'))
print(get_nutrient(nutrient, 'บัวลอย เผือก'))
print(get_nutrient(nutrient, 'ข้าว กะเพรา ไก่'))
print(get_nutrient(nutrient, 'ลาไย เผือก'))
```

จะได้ผลลัพธ์เป็น

```
[]
[336.0, 62.0, 2.0, 9.0, 'NA', 'NA']
[445.0, 57.0, 20.0, 15.0, 'NA', 1200.0]
['NA', 'NA', 'NA', 'NA', 'NA', 'NA']
```

### งานที่ 4 สรุปสารอาหารที่บริโภคประจำวัน

ผู้บริโภคจะเก็บบันทึกรายการการรับประทานอาหารของตน (intakes) ในลักษณะเป็นลิสต์ของลิสต์ในรูปแบบ

แบบเป็น วันที่ ตามด้วย ชื่ออาหาร 1 ชื่อ เรียงต่อกันยาวเท่าไรก็ได้ อาจจะบันทึกวันละหลายรายการได้ หรือ บันทึกย้อนหลังได้ หรืออาจจะไม่บันทึกการรับประทานบางวันก็ได้ ตัวอย่างของลิสต์ intakes เป็นดังนี้

```
[['2022/01/15', 'ลาบ หมู'],
 ['2022/01/15', 'ส้มตำ ไทย'],
 ['2022/01/18', 'กาแฟ เย็น'],
 ['2022/01/14', 'แซนวิช'],
 ['2022/01/19', 'ชาวน้ำ'],
 ['2022/01/15', 'ขนมจีน น้ำยา'],
 ['2022/01/16', 'ขนมจีน น้ำพริก'],
 ['2022/01/16', 'ส้มตำ ไทย'],
 ['2022/01/16', 'สลัด ทูน่า'],
 ['2022/01/18', 'ข้าว ไก่ ผัก กะเพรา'],
 ['2022/01/18', 'ส้มตำ ไทย'],
 ['2022/01/18', 'แตงโม'],
 ['2022/01/14', 'ส้มตำ ไทย'],
 ['2022/01/14', 'ลาบ หมู'],
 ['2022/01/19', 'บัวลอย']]
```

ให้เขียนฟังก์ชัน `summarize_daily_intake(nutrient, intakes)` ที่

- รับ `nutrient` เก็บตารางข้อมูลโภชนาการรูปแบบของลิสต์ที่อธิบายข้างบน
- รับ `intakes` เก็บรายการการรับประทานอาหารของผู้ใช้ ในรูปแบบของลิสต์ที่อธิบายข้างต้น
- คืน ลิสต์ผลลัพธ์ที่เก็บข้อมูลสรุปโภชนาการสารอาหารที่ได้รับในแต่ละวัน ในรูปแบบของลิสต์ของลิสต์รูปแบบเป็น `[date, dailykcal, dailycarb, dailyprotein, dailyfat, dailysugar, dailysodium, complete]` โดยที่
  - `dailykcal, dailycarb, dailyprotein, dailyfat, dailysugar, dailysodium` คือ ยอดรวมของ ค่าพลังงาน(kcal) ปริมาณ คาร์โบไฮเดรต(g) โปรตีน(g) ไขมัน(g) น้ำตาล(g) โซเดียม(mg) ของทุกอาหารที่บริโภคในวันนั้น ๆ
  - `complete` เป็นค่า Boolean โดยให้ค่าเริ่มต้นเป็น `True`
  - ในกรณีที่ ชื่ออาหารที่บริโภค ไม่คล้ายกับชื่ออาหารในตารางโภชนาการ `nutrient` (`match_foods()` คืนค่าลิสต์ว่าง) จะไม่นำอาหารนี้มาคำนวณ และให้ `complete` มีค่าเป็น `False`
  - ในกรณีที่ ชื่ออาหารที่บริโภค คล้ายกับชื่ออาหารในตารางโภชนาการ `nutrient` มากกว่า 1 รายการ ให้ใช้ ค่าเฉลี่ย เหมือนวิธีคำนวณใน `get_nutrient()`
  - ปริมาณสารอาหารที่มีค่าเป็น `NA` จะไม่นำมาคิดเป็นส่วนหนึ่งของค่าเฉลี่ย
  - ลิสต์ผลลัพธ์ต้องเรียงลำดับวันที่จากวันที่ล่าสุดไปอดีต

ตัวอย่างเช่น หากเรียกใช้ฟังก์ชันนี้โดยคำสั่ง

```
intake_summary = summarize_daily_intake(nutrient, intakes)
for item in intake_summary:
    print(item)
```

จะได้ผลลัพธ์เป็น

```
['2022/01/19', 773.0, 124.0, 11.0, 26.0, 'NA', 810.0, True]
['2022/01/18', 575.0, 197.0, 51.0, 20.0, 27.0, 1064.0, False]
['2022/01/16', 640.0, 218.0, 42.0, 22.0, 27.0, 1064.0, False]
['2022/01/15', 758.0, 201.0, 68.0, 31.0, 40.0, 3744.0, True]
['2022/01/14', 410.0, 160.0, 54.0, 17.0, 29.0, 2534.0, False]
```

## ▼ งานของนิสิต

จงเขียนฟังก์ชัน `compute_similarity()`, `match_foods()`, `get_nutrient()` และ `summarize_daily_intake()` ใน code cell ด้านล่างนี้

### การส่งงาน

- Save a copy in Drive
- เปลี่ยนชื่อ Notebook ให้เป็น HW6\_XXXXXXXXXX.ipynb โดย XXXXXXXXXXXX เป็นเลขประจำตัวนิสิต
- File -> Download -> Download .ipynb
- ส่งไฟล์ที่ดาวน์โหลดได้ใน MyCourseVille ก่อนเวลา 23:59 น. วันที่ 23 กุมภาพันธ์ 2565
- จะตรวจแฟ้ม .ipynb แฟ้มล่าสุดที่ส่งในระบบ MyCourseVille เท่านั้น

### สำคัญ: อ่านตรงนี้ด้วย

- ห้ามเปลี่ยนบรรทัดแรกของ code cell
- ห้ามเปลี่ยนบรรทัด def ของฟังก์ชันที่ให้เขียน
- ฟังก์ชันต้องคืนข้อมูลและประเภทข้อมูลตามที่กำหนดเท่านั้น
- ฟังก์ชันต้องไม่ใช้ข้อมูลในตัวแปรใด ๆ นอกฟังก์ชัน
- ฟังก์ชันต้องไม่เปลี่ยนแปลงข้อมูลภายในของพารามิเตอร์ที่ได้รับ (หากระบุไว้ในข้อกำหนด)
- ไม่อนุญาตให้ใช้ dict, set
- ส่งแฟ้ม .ipynb ก็ครั้งก็ได้ แต่จะตรวจแฟ้ม .ipynb แฟ้มล่าสุด เท่านั้น
- โปรแกรมที่ทำงานไม่ได้ เกิด error ก็ไม่ได้คะแนน
- ห้าม import อะไรใด ๆ เพิ่ม (แม้จะ import แต่ไม่ใช้ก็ตาม) ห้ามใช้ dict กับ set นอกนั้น อยากรู้ค่าสิ่งอะไร ก็ใช้ได้ ถ้าทำงานได้ใน [Python version 3.7](#)
- **ข้อทุจริต** เช่น
  - ส่งโปรแกรมที่ผู้ส่งไม่สามารถอธิบายได้ว่า ใช้หลักการและทำงานอย่างไร
  - หรือ ส่งโปรแกรมที่คล้ายกับโปรแกรมของผู้อื่นมาก ๆ (ไม่ว่าจะเป็นผู้ให้หรือผู้รับ จะตั้งใจหรือไม่ก็ตาม)
- หากพบว่า งานที่ส่งข้อทุจริต จะได้คะแนน **ติดลบคะแนนรวมสะสมตั้งแต่การบ้านครั้งแรกสุด**

```

# HW6_LIST (ไม่ลบหรือแก้ไขบรรทัดนี้ หรือเพิ่มอะไรก่อนบรรทัดนี้ โดยเด็ดขาด)

# - เขียนในเซลล์นี้เท่านั้น
# - ถ้าต้องการเขียนฟังก์ชันเพิ่ม ก็เขียนในเซลล์นี้

def compute_similarity(food_name1, food_name2):
    """
    รับ food_name1 และ food_name2 เป็นสตริงเก็บชื่ออาหารที่ตัดคำมาให้แล้ว โดยมีวรรคคั่นระหว่างคำ
    คืน ค่าความคล้ายของชื่ออาหาร ซึ่งเป็นเลขจำนวนจริงที่มีค่าระหว่าง [0,1]
    """

# -----
def match_foods(nutrient, food_name):
    """
    รับ nutrient เก็บตารางข้อมูลโภชนาการรูปแบบของลิสต์ที่อธิบายข้างบน
    รับ food_name เป็นสตริงชื่ออาหารที่ผู้บริโภครับประทานที่ตัดคำมาให้แล้ว โดยมีวรรคคั่นระหว่างคำ
    คืน ลิสต์ผลลัพธ์ที่เก็บรหัสอาหาร food_id ที่มีชื่ออาหารตรงกับ food_name มากที่สุด รายละเอียดดังอธิบายไว้ข้างบน
    """

# -----
def get_nutrient(nutrient, food_name):
    """
    รับ nutrient เก็บตารางข้อมูลโภชนาการ ในรูปแบบของลิสต์
    รับ food_name เป็นสตริงชื่ออาหาร
    คืน ลิสต์ผลลัพธ์ที่เก็บข้อมูลพลังงานและสารอาหาร รายละเอียดดังอธิบายไว้ข้างบน
    """

# -----
def summarize_daily_intake(nutrient, intakes):
    """
    รับ nutrient เก็บตารางข้อมูลโภชนาการ ในรูปแบบของลิสต์ที่อธิบายข้างต้น
    รับ intakes เก็บรายการการรับประทานอาหารของผู้ใช้ ในรูปแบบของลิสต์ที่อธิบายข้างต้น
    คืน ลิสต์ผลลัพธ์ที่เก็บข้อมูลสรุปโภชนาการสารอาหารที่ได้รับในแต่ละวัน รายละเอียดตามที่อธิบายข้างบน
    """

# -----
def main():

    nutrient = [['R010007', 'ลาบ หมู', 267, 17, 23, 12, 2, 1470],
                 ['R010014', 'ส้มตำ ไทย', 143, 143, 31, 5, 27, 1064],
                 ['R010005', 'ก๋วยเตี๋ยว ผัดไทย ใส่ ไข่', 447, 49, 21, 18, 'NA', 1139],
                 ['P010019', 'ขนมจีน ชวน้ำ', 437, 62, 9, 17, 'NA', 810],

```

```
[
    ['P010021', 'ขนมจีน น้ำยา', 348, 41, 14, 14, 11, 1210],
    ['P010023', 'ขนมจีน น้ำพริก', 497, 75, 11, 17, 'NA', 'NA'],
    ['P010041', 'ข้าว หมก ไก่', 481, 74, 19, 12, 'NA', 900],
    ['P020008', 'ข้าว ราด กะเพรา ไก่', 458, 60, 20, 15, 'NA', 1200],
    ['P010049', 'ข้าว ไข่ พะโล้', 464, 55, 20, 18, 'NA', 946],
    ['P010025', 'ข้าว ไก่ ผัด กะเพรา', 432, 54, 20, 15, 'NA', 'NA'],
    ['F010003', 'กล้วย ไข่', 62, 14, 0, 0, 7, 4],
    ['D010032', 'บัวลอย เผือก', 336, 62, 2, 9, 'NA', 'NA'],
    ['D010033', 'ลาบ ไข่', 'NA', 'NA', 'NA', 'NA', 'NA', 'NA']
]
```

```
intakes = [
    ['2022/01/15', 'ลาบ หมู'],
    ['2022/01/15', 'ส้มตำ ไทย'],
    ['2022/01/18', 'กาแฟ เย็น'],
    ['2022/01/14', 'แซนวิช'],
    ['2022/01/19', 'ข้าวน้ำ'],
    ['2022/01/15', 'ขนมจีน น้ำยา'],
    ['2022/01/16', 'ขนมจีน น้ำพริก'],
    ['2022/01/16', 'ส้มตำ ไทย'],
    ['2022/01/16', 'สลัด ทูน่า'],
    ['2022/01/18', 'ข้าว ไก่ ผัด กะเพรา'],
    ['2022/01/18', 'ส้มตำ ไทย'],
    ['2022/01/18', 'แดงโม'],
    ['2022/01/14', 'ส้มตำ ไทย'],
    ['2022/01/14', 'ลาบ หมู'],
    ['2022/01/19', 'บัวลอย']
]
```

```
print(compute_similarity('ข้าว กะเพรา ไก่', 'ข้าว ไก่ ผัด กะเพรา'))
print(match_foods(nutrient, 'ข้าว กะเพรา ไก่'))
print(get_nutrient(nutrient, 'ห่อ หมก'))
print(summarize_daily_intake(nutrient, intakes))
```

```
main()
```

