

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....
กำหนดส่ง



กระบวนวิชา **229223**

Lab	
HW	
Until	

การบ้านปฏิบัติการ 11

Collections and I/O Redirections (20 คะแนน)

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ `import` ไปเรียกใช้งานจาก Script อื่น ๆ ได้
- นักศึกษาสามารถสร้างฟังก์ชันย่อยต่าง ๆ เพิ่มเติมได้ตามความเหมาะสม และเลือกใช้การแก้ปัญหาได้ทั้งวิธี `iteration` หรือ `recursion`

- 4 คะแนน (Lab11_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `word_count(text)` เพื่อคืนค่า dictionary ผลลัพธ์จากการนับจำนวนคำที่ปรากฏในสายอักขระ (String) `text` โดยฟังก์ชันจะ คืนค่าเป็น dict ที่มี `key` เป็นแต่ละคำที่ปรากฏใน `text` และมี `value` เป็นความถี่ ทั้งนี้ตัวอักษรที่อยู่ใน `key` จะต้องเป็นตัวอักษรพิมพ์เล็กเท่านั้น

ข้อกำหนด

- การนับความถี่จะเป็นแบบ Case Insensitive ('ant' และ 'Ant' ถือเป็นคำเดียวกัน)
- ข้อความในไฟล์จะเป็นภาษาอังกฤษมาตรฐานในรูปแบบที่ถูกต้อง (well-formed English)
- ไม่พิจารณาเครื่องหมายวรรคตอนต่าง ๆ เฉพาะที่ล้อมรอบคำ เช่น `!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~`
- เนื่องจาก `key` มีคุณสมบัติเป็น `set` ลำดับในการแสดงผลใน output จึงไม่จำเป็นต้องเหมือนตัวอย่าง

Input

Output

"He doesn't want to pay \$40,000 for a new car, but his wife doesn't seem to care."	{ 'new': 1, 'but': 1, 'pay': 1, 'want': 1, 'seem': 1, 'care': 1, 'his': 1, '40,000': 1, 'wife': 1, 'a': 1, 'for': 1, 'car': 1, "doesn't": 2, 'to': 2, 'he': 1}
---	--

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

• การวิเคราะห์ปัญหา

- Input: จำนวนข้อมูล _____ ชนิดข้อมูล _____
- Output: (แสดงค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____
(คืนค่า) จำนวนข้อมูล _____ ชนิดข้อมูล _____

2) 4 คะแนน (Lab11_2_6XXXXXXX.py) [Attachments] ลูฟี่ ตัวละครหลักจากอนิเมะเรื่อง One Piece ได้พบสมุดบันทึกเก่าแก่ซึ่งมีตำแหน่งของสมบัติทั้งหมดในโลกของ One Piece ในสมุดที่พบมีข้อมูลเกี่ยวกับตำแหน่งของสมบัติ (location) ประเภทของสมบัติ (treasure type) และมูลค่าของสมบัติ (value) ลูฟี่และสมาชิกกลุ่มหมวกฟางต้องการทราบมูลค่ารวมของสมบัติประเภทใดประเภทหนึ่งเช่น มูลค่ารวมของทองจากทุกจุดในแผนที่หน้าที่คุณจะต้อง เขียนฟังก์ชันดังต่อไปนี้:

a) ฟังก์ชัน `read_input()` เพื่ออ่านข้อมูลสมบัติจากสมุดบันทึกด้วยวิธี command redirection และคืนค่า **dict** ที่มี **key** เป็นประเภทของสมบัติ โดยไฟล์หนึ่งๆ จะแทน 1 หน้าของสมุดบันทึกซึ่งมีรูปแบบดังนี้

- บรรทัดแรกสุดจะขึ้นต้นด้วยอักขระ '#' เสมอและแสดงข้อความแทนเลขหน้าของสมุดบันทึก และข้อมูลอื่น ๆ ซึ่งฟังก์ชันจะต้องไม่นำบรรทัดนี้มาใช้คำนวณ

- บรรทัดถัดๆ ไปมีรูปแบบดังนี้:

<location>, <treasure_type>, <value>

- จะไม่มีบรรทัดไหนที่เหมือนกันทั้งตำแหน่ง (location) และ ประเภทของสมบัติ (treasure type)

โดย **dict** ที่คืนค่าจะมีรูปแบบคือ มี **key** เป็น treasure_type (ประเภทของสมบัติที่พบ) และ **value** เป็น list ของ tuple ในรูปของ (location, value) หรือ (ตำแหน่ง, มูลค่า)

Input: 'page28.txt'

Output:

# page 28 - year 1400 Marineford, Platinum, 42000 Marineford, Diamond, 300 Enies Lobby, Bronze, 43000 Elbaf, Silver, 9600 Dressrosa, Gold, 190 Punk Hazard, Gold, 900	{ "Platinum": [("Marineford", 42000)], "Diamond": [("Marineford", 300)], "Bronze": [("Enies Lobby", 43000)], "Silver": [("Elbaf", 9600)], "Gold": [("Dressrosa", 190), ("Punk Hazard", 900)] }
---	--

b) ฟังก์ชัน `total_value(treasure_type, treasures)` เพื่อคืนค่ามูลค่ารวมที่คำนวณตามประเภทของสมบัติที่ระบุด้วยตัวแปร `treasure_type` เมื่อ `treasures` คือ **dict** ที่ได้จาก output ของฟังก์ชัน `read_input()` และฟังก์ชันจะคืนค่า -1 หากไม่พบประเภทของสมบัติตามที่ระบุด้วยตัวแปร `treasure_type`

Function Call

Output:

<code>total_value('Gold', treasures)</code>	1090
<code>total_value('Ruby', treasures)</code>	-1

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

- 3) 4 คะแนน (HW11_1_6XXXXXXX.py) ให้เขียนฟังก์ชัน `display_calendar(month, year)` เพื่อแสดงปฏิทินในเดือนและปีคริสต์ศักราชตามปฏิทินกริกอเรียน (Gregorian Calendar) ที่ระบุด้วยตัวแปร `month` และ `year` โดยให้ศึกษา Algorithm การคำนวณวันในสัปดาห์ (อาทิตย์, จันทร์, อังคาร, ... เสาร์) เมื่อทราบวันเดือนปีจาก

https://en.wikipedia.org/wiki/Zeller%27s_congruence#Formula ทั้งนี้ไม่อนุญาตให้ใช้ module `datetime` ในการแก้ปัญหา

ข้อกำหนด: หลังแสดงผลวันสุดท้ายของเดือนฟังก์ชันจะต้องแสดงอักขระ new line ('\n') ทันที

Function Call

`display_calendar(2, 2023)`

Output

Su Mo Tu We Th Fr Sa
 1 2 3 4 5 6 7
 8 9 10 11 12 13 14
 15 16 17 18 19 20 21
 22 23 24 25 26 27 28

Hint: สามารถใช้ คำสั่ง `cal` ใน bash shell (ตัวอย่างเช่น <https://replit.com/languages/bash>) เพื่อตรวจสอบ output ที่ถูกต้อง เช่น `cal 2 2023`

- การวิเคราะห์ปัญหา

- Input:

จำนวนข้อมูล

ชนิดข้อมูล

- Output:

(แสดงค่า)

จำนวนข้อมูล

ชนิดข้อมูล

(คืนค่า)

จำนวนข้อมูล

ชนิดข้อมูล

- 4) 4 คะแนน (HW11_2_5XXXXXXX.py) ให้เขียนให้เขียนฟังก์ชัน `runner_up()` เพื่อรับค่าคะแนนของนักศึกษาในห้อง ตามจำนวนนักศึกษาที่ระบุในบรรทัดแรก แล้วแสดงผลคะแนนที่สูงเป็นอันดับที่ 1, อันดับที่ 2 และ ค่าเฉลี่ยคะแนน (ทศนิยม 2 ตำแหน่ง) โดยหากไม่มีตำแหน่งที่ 2 ให้ แสดงคำว่า `None`

ข้อกำหนด: ไม่อนุญาตให้ใช้ตัวแปรประเภท `iterables` เช่น `list` หรือ `tuple` ในการเก็บข้อมูลที่ได้รับเพื่อการคำนวณ

Hint: ศึกษา Slide Input/Output Redirection เพื่อความสะดวกในการทดสอบข้อมูลนำเข้าจำนวนมาก

ตัวอย่างการ run 1

Total students: 3
 Enter score:
 13
 12
 13

 Max score is: 13.00
 Runner up is: 12.00
 Average is: 12.67

ตัวอย่างการ run 2

Total students: 2
 Enter score:
 61
 61

 Max score is: 61.00
 Runner up is: None
 Average is: 61.00

ชื่อ-นามสกุล.....รหัสนักศึกษา.....ตอนเรียน.....ลำดับที่.....

ตัวอย่างการ run 3

Total students: 7

Enter score:

61

72

64

81

61

79

63

Max score is: 81.00

Runner up is: 79.00

Average is: 68.71

ตัวอย่างการ run 4

Total students: 1

Enter score:

13

Max score is: 13.00

Runner up is: None

Average is: 13.00

- 5) 4 คะแนน (HW11_3_6XXXXXXXXX.py) ให้เขียนฟังก์ชัน `words_to_num(words)` เพื่อทำการคืนค่าจำนวนเต็มบวกที่คำนวณได้จากคำอ่านของจำนวนเต็ม `words` โดยผลลัพธ์จะมีความยาวไม่เกิน 12 หลัก

Hint: สามารถศึกษาการอ่านตัวเลขในภาษาอังกฤษได้จาก http://en.wikipedia.org/wiki/English_numerals

Input

Output

fourteen	14
two hundred forty-eight	248
one hundred eleven	111
forty-two billion six hundred forty-one million three hundred twenty-three thousand eight hundred sixty-two	42641323862

- การวิเคราะห์ปัญหา

- Input:

จำนวนข้อมูล

ชนิดข้อมูล

- Output:

(แสดงค่า)

จำนวนข้อมูล

ชนิดข้อมูล

(คืนค่า)

จำนวนข้อมูล

ชนิดข้อมูล

การส่งงาน

1. ลักษณะ/ลำดับข้อความของการรับค่า/แสดงผล จะต้องเป็นไปตามที่ระบุในตัวอย่างการ run
2. ไฟล์งานที่ส่ง จะต้องมีการแทรก comment ที่ต้นไฟล์ตามข้อกำหนดใน canvas รายวิชา
3. ไฟล์งานโปรแกรมที่ส่ง จะต้องมีการแทรก pseudocode เป็น comment ในแต่ละขั้นตอน
4. Upload ไฟล์ source code ตามที่ระบุในแต่ละข้อ ไปยังระบบตรวจให้คะแนนอัตโนมัติ <https://cmu.to/gdr223>