



Assignment 2- Extension



作業檔案

如果同學完成Assignment2，歡迎撰寫這份Extension作業，強化各位同學運用Python 程式最重要的基本技能：基本的數學運算式 (Expressions)、與使用者互動之對畫框 (Console)、以及各式迴圈 (Loops)。

如果作業卡關 **歡迎與助教討論**，stanCode也非常鼓勵同學們互相討論作業的概念，**但請不要把自己的code給任何人看**，分享您的code會剝奪其他學生獨立思考的機會，同時會導致其他學生的程式碼與您的code極度相似，使得防抄襲軟體認定有抄襲之嫌疑。

如果有做這份 extension，請於完成後，將extension的py檔直接複製貼上於原作業資料夾內。

Problem 1 - factorial.py

stanCode接獲神秘計算機協會的請求，想請各位同學協助完成一個可以把得到的數字相乘的程式。請同學完成 **factorial.py**，寫出一個可以讓使用者反覆輸入數字，並將這些輸入的數字相乘的程式。如下圖所示，您所完成的程式應該要能完美重現下圖內所有文字與數字：

```
Welcome to stanCode factorial master!
Give me a number, and I will list the answer of factorial: 1
Answer: 1
Give me a number, and I will list the answer of factorial: 2
Answer: 2
Give me a number, and I will list the answer of factorial: 3
Answer: 6
Give me a number, and I will list the answer of factorial: 4
Answer: 24
Give me a number, and I will list the answer of factorial: 5
Answer: 120
Give me a number, and I will list the answer of factorial: 6
Answer: 720
Give me a number, and I will list the answer of factorial: 7
Answer: 5040
Give me a number, and I will list the answer of factorial: 8
Answer: 40320
Give me a number, and I will list the answer of factorial: 9
Answer: 362880
Give me a number, and I will list the answer of factorial: 10
Answer: 3628800
Give me a number, and I will list the answer of factorial: -100
----- See ya! -----
```

以下兩點請同學注意：

1. 一開始請印出 'Welcome to stanCode factorial master!'
2. 最後，請在main function上方，設定一個讓使用者可以輸入「使程式離開的值」，當使用者輸入這個EXIT常數時（如上圖中的 -100），那程式就要印出“----- See ya! -----”並結束程式。

Problem 2 - number_checker.py

今天如果我們把一個整數所有的真因子(即除了自身以外的因數)全部加起來，

1. **真因子的和** 如果**等於**這一個**整數**，則這個整數就可以被稱為**“完全數”**
(Perfect number)。
2. **真因子的和**如果**大於**這一個**整數**，則我們會將這個整數稱為**“盈數”**
(abundant number)。
3. 最後，**真因子的和**如果**小於**這一個**整數**，則我們會將這個整數稱為**“虧數”**
(deficient number)。

舉例來說，

完全數 (Perfect number) 的例子有 **6** 跟 **28**：

- 數字 **6** 的因數有 **1、2、3、6**，扣除 **6** 本身， $1+2+3 == 6$ ，因此 **6** 為完美數。
- 數字 **28** 的因數有 **1、2、4、7、14、28**，除去數字 **28** 本身外，其餘5個數相加， $1+2+4+7+14 == 28$ ，因此 **28** 為完美數。

盈數 (abundant number) 的例子有 **12** 跟 **20**：

- 數字 **12** 的因數有 **1、2、3、4、6、12**，扣除 **12** 本身， $1+2+3+4+6 == 16$ ，16 大於 12，因此 **12** 為盈數。
- 數字 **20** 的因數有 **1、2、4、5、10、20**，將除去數字 **20** 自己以外的五個數字加起來， $1+2+4+5+10 == 22$ 。22 大於 20，因此 **20** 為盈數。

虧數 (deficient number) 的例子有 **15** 跟 **27**：

- 數字 **15** 的因數有 **1、3、5、15**，扣除 **15** 本身， $1+3+5 == 9$ ，9 小於 15，因此 **15** 為虧數。
- 數字 **27** 的因數有 **1、2、3、9、27**，將除去數字 **27** 自己以外的四個數字相加， $1+2+3+9 == 15$ 。15 小於 27，因此 **27** 為虧數。

請同學完成 `number_checker.py`，寫出一個讓使用者可以反覆輸入數字並判斷該數字為完全數、盈數還是虧數的程式。

如下圖所示，您所完成的程式應該能完美重現圖中的每一行文字與數字。

```
Welcome to the number checker!
```

```
n: 6
```

```
6 is a perfect number
```

```
n: 15
```

```
15 is a deficient number
```

```
n: 8128
```

```
8128 is a perfect number
```

```
n: 12
```

```
12 is an abundant number
```

```
n: 79
```

```
79 is a deficient number
```

```
n: 945
```

```
945 is an abundant number
```

```
n: -100
```

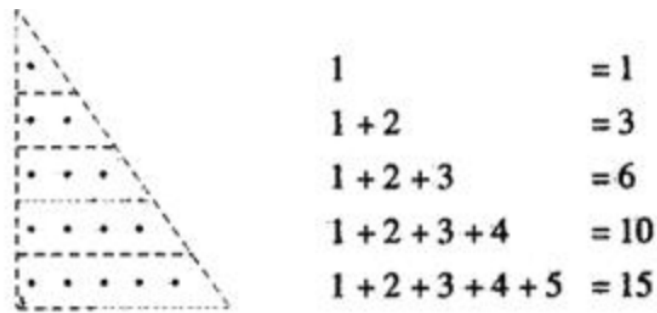
```
Have a good one!
```

以下三點請同學注意：

1. 一開始請印出 'Welcome to the number checker!'
2. 您可以假設使用者在 "n:" 後面輸入的數字一定是大於 1 的正整數，不需要做任何 error checking
3. 最後，請在main function上方，設定一個讓使用者可以輸入「使程式離開的值」，當使用者輸入這個EXIT常數時（如上圖中的 -100），那程式就要印出 "Have a good one!" 並結束。

Problem 3 - triangular_checker.py

如果一個整數，我們拿跟這個整數一樣多的石頭排在地上，如果在等距離的排列下可以形成一個等邊三角形(如下圖)，則這個數就可以被視為 “**三角形數**”(triangular number)。



我們可以透過下列公式來找到三角形數， n 代表是第 n 個三角形數。

$$T_n = n(n + 1)/2$$

舉例來說，

- **3** 是第 **2** 個三角形數，因為他符合： $2(2+1) / 2 == 3$
- **15** 是第 **5** 個三角形數，因為他符合： $5(5+1) / 2 == 15$
- **28** 是 **7** 個三角形數，因為他符合： $7(7+1) / 2 == 28$

請同學完成 `triangular_checker.py`，寫出一個讓使用者可以反覆輸入數字並判斷該數字是否為三角形數的程式。

如下圖所示，您所完成的程式應該能完美重現圖中的每一行文字與數字。

```
Welcome to the triangular number checker!
n: 3
3 is a triangular number
n: 17
17 is not a triangular number
n: 171
171 is a triangular number
n: 12
12 is not a triangular number
n: 21
21 is a triangular number
n: 28
28 is a triangular number
n: 40
40 is not a triangular number
n: 371
371 is not a triangular number
n: -100
Have a good one!
```

以下兩點請同學注意：

1. 一開始請印出 'Welcome to the triangular number checker!'
2. 最後，請在main function上方，設定一個讓使用者可以輸入「使程式離開的值」，當使用者輸入這個EXIT常數時（如上圖中的 -100），那程式就要印出 "Have a good one!" 並結束。

Problem 4 - narcissistic_checker.py

在數學理論的世界中，如果一個**正整數**的**所有位數**的**N次方**的**和**剛好**等於自己**，那他就會被稱為 **水仙花數 (Narcissistic number)** ！

水仙花數 (Narcissistic number)，又稱超完全數字不變數 (pluperfect digital invariant, PPDi)、自戀數、自冪數、阿姆斯壯數 (Armstrong number)，之所以稱為阿姆斯壯數，是因為據說這個概念最早是由阿姆斯壯提出的！

舉例來說：

- $153 == 1^3 + 5^3 + 3^3$ ，因此 153 是水仙花數
- $1634 == 1^4 + 6^4 + 3^4 + 4^4$ ，因此 1634 是水仙花數
- $92727 == 9^5 + 2^5 + 7^5 + 2^5 + 7^5$ ，因此 92727 是水仙花數

請同學完成 **narcissistic_checker.py**，寫出一個讓使用者可以反覆輸入數字並判斷該數字是否為水仙花數的程式。如下圖所示，您所完成的程式應該要能**完美重現**下圖內所有文字與數字：

```
Welcome to the narcissistic checker!
n: 153
153 is a narcissistic number
n: 1634
1634 is a narcissistic number
n: 92727
92727 is a narcissistic number
n: 120
120 is not a narcissistic number
n: 11
11 is not a narcissistic number
n: -100
Have a good one!
```

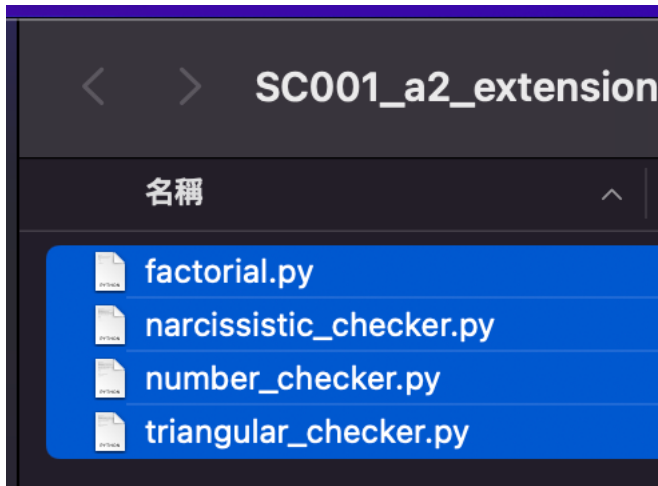
以下三點請同學注意：

1. 一開始請印出 'Welcome to the narcissistic number checker!'
2. 最後，請在main function上方，設定一個讓使用者可以輸入「使程式離開的值」，當使用者輸入這個EXIT常數時（如上圖中的 -100），那程式就要印出 "Have a good one!" 並結束。
3. 提示：這題建議使用 // 與 % ！

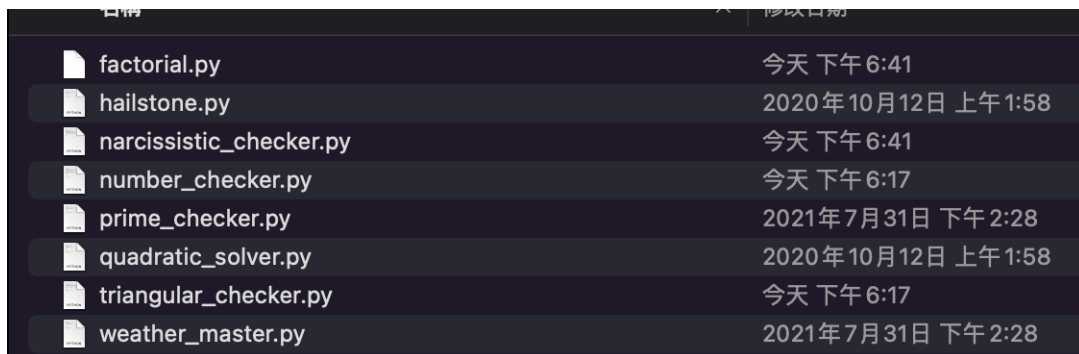
作業繳交

恭喜您完成Assignment2 Extension！請同學於作業繳交期限前，依照下圖將您完成的作業的下載連結上傳至社團提供的作業繳交表單。

1. 將 SC001_a2_extension資料夾內的檔案全選，使用 **command(ctrl)+c** 複製

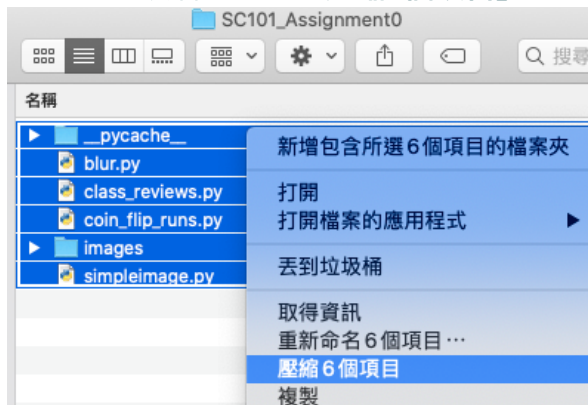


2. 將複製的 py檔案，至assignment2作業資料夾內，使用 **command(ctrl) + v** 貼上檔案。

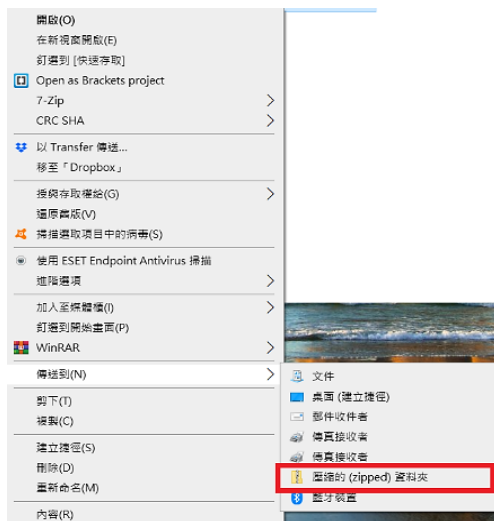


3. 以滑鼠「全選」作業資料夾內的所有檔案，並壓縮檔案。請見下圖說明。

macOS：按右鍵選擇「壓縮n個項目」



Windows：按右鍵選擇「傳送到」→「壓縮的(zipped)資料夾」



4. 將壓縮檔(.zip)重新命名為「a2_中文姓名」。如：



5. 將命名好的壓縮檔(.zip)上傳至Google Drive (或任何雲端空間)

- 1) 搜尋「google drive」
- 2) 登入後，點選左上角「新增」→「檔案上傳」→選擇作業壓縮檔(.zip)

6. 開啟連結共用設定，並複製下載連結

- 1) 對檔案按右鍵，點選「共用」
- 2) 點擊「變更任何知道這個連結的使用者權限」後，權限會變為「可檢視」
- 3) 點選「複製連結」



7. 將連結上傳至作業貼文提供的「作業提交表單」