



Assignment 2

This assignment is based on the Assignment 2 of CS106A at Stanford University



作業檔案下載

這份作業將訓練各位同學 `python` 程式最重要的基本技能：完成基本的數學運算式 (Expressions)、與使用者互動之對畫框 (Console)、以及各式迴圈 (Loops)

估計需要時間為 6 小時

繳交期限 – 2020/10/22 (四) 23:59

如果作業卡關歡迎各位到社團提問，也非常鼓勵同學們互相討論作業之概念，但請勿把 `code` 給任何人看（也不要將程式碼貼在社團裡）分享妳/你的 `code` 會剝奪其他學生獨立思考的機會，也會因此讓其他學生的程式碼與妳/妳的極度相似，讓防抄襲軟體認定有抄襲嫌疑

如果真的有 `code` 方面的問題，可以在計概課堂下課詢問或是利用 TA 時間詢問 Jerry 或助教。只有他們可以看您的 `code`

Problem 1 - quadratic_solver.py

假設我們有一個二次函數如下圖方程式所示（**a 不等於零**）

$$ax^2+bx+c=0$$

只要使用者給定 **a, b, c** 三個數值，我們就可以依照下方公式計算出此方程式的 **roots** (也就是下方 **x** 的值)

$$X=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

在根號裡面的 **b² - 4ac** 我們稱之為 **discriminant**：如果大於零，此方程式會有兩個 **real roots** (兩個 **x** 的值)，同上方算式的兩組數值

然而，如果 **discriminant** 等於零，我們只會有一個 **root** (一個 **x** 的值)

如果 **discriminant** 小於零，我們要告訴使用者 "No real roots"

請寫出一個可以重建下方三個圖例之程式 (由左至右，分別代表 **2 real roots**, **1 root**, 以及 **No real roots**。要產生這三張圖必須將程式關掉重新啟動 (recompile) 三次)

首先，請告訴使用者此程式名稱 ("stanCode Quadratic Solver!") 然後分別提供三行文字讓使用者輸入 **a, b, c** 相對應之數值

若您要計算開根號 (√) 請使用指令 **math.sqrt**。舉例來說，若我們要將 **x** 開根號的數值存到一個 **y**:

$$y = \text{math.sqrt} (x)$$

math.sqrt 來自我們程式最上方的 **import math**，同學們請先不用擔心直接使用

```
stanCode Quadratic Solver!  
Enter a: 1  
Enter b: -3  
Enter c: -4  
Two roots: 4.0 , -1.0
```

```
stanCode Quadratic Solver!  
Enter a: 1  
Enter b: 6  
Enter c: 9  
One root: -3.0
```

```
stanCode Quadratic Solver!  
Enter a: 2  
Enter b: 4  
Enter c: 6  
No real roots
```

Problem 2 - weather_master.py

中央氣象局請同學幫忙處理天氣資料，身為 **stanCode** 學生的我們，當然就要使用程式替我們工作囉（挺）

中央氣象局希望我們特別注意在所有輸入程式數據中的四個數值：**最高溫**是多少？**最低溫**是多少？**平均溫度**是多少？以及有幾天可以發佈「**低溫警報**」（小於 16 度但不包含 16 度）所以聰明的 妳/你 就想到，何不使用上課所教的程式概念來處理這個問題？

您的程式會反覆請使用者輸入一個整數；若使用者想要離開程式只要輸入 -100 即可。然而，身為一個好的 **programmer**，我們要將「使程式離開的值（-100）」存入一個位於 **main()** 上方的 **constant**，就好像終極密碼最終的數字一樣。您寫的程式應該要可以得出與下圖一模一樣的內容：

```
stanCode "Weather Master 4.0"!  
Next Temperature: (or -100 to quit)? 20  
Next Temperature: (or -100 to quit)? 16  
Next Temperature: (or -100 to quit)? 8  
Next Temperature: (or -100 to quit)? 13  
Next Temperature: (or -100 to quit)? 19  
Next Temperature: (or -100 to quit)? 24  
Next Temperature: (or -100 to quit)? 33  
Next Temperature: (or -100 to quit)? 31  
Next Temperature: (or -100 to quit)? -100  
Highest temperature = 33  
Lowest temperature = 8  
Average = 20.5  
2 cold day(s)
```

眼尖的同学一定有注意到，在 `print ("")` 括弧的双引号内再填入双引号是不可能的事。举例来说，`print("SC001 "Weather")` 会让电脑认为我们只需要 `"SC001 "`

为了解决这件事情，电脑工程师就把「当 `\` 出现在双引号 (`""`) 内」定义为显示引号的方法。举例来说，`print("SC001 \" Weather")` 就会印出 **SC001 "Weather**

如同题幹一开始的叙述，身为一个好的 programmer，我们要将「使程式离开值 (-100)」存入一个位于 `weather()` 上方的 constant。然而，如果我们更改它的值，从 -100 变成 -1 并重跑 (recompile / run) 程式，画面就会变成：

```
stanCode "Weather Master 4.0"!
Next Temperature: (or -1 to quit)? 33
Next Temperature: (or -1 to quit)? 22
Next Temperature: (or -1 to quit)? 27
Next Temperature: (or -1 to quit)? -1
Highest temperature = 33
Lowest temperature = 22
Average = 27.333333333333332
0 cold day(s)
```

请同学注意的地方是，使用者可以只输入一个数值（如下图，当我们只输入 3）而不影响我们判别最高温、最低温、以及平均气温（应该都要是 3）

```
stanCode "Weather Master 4.0"!
Next Temperature: (or -1 to quit)? 3
Next Temperature: (or -1 to quit)? -1
Highest temperature = 3
Lowest temperature = 3
Average = 3.0
1 cold day(s)
```

最后，如果使用者一开始便输入「使程式离开值」，那我们就要印出

No temperature were entered. 的字样，如下图所示：

```
stanCode "Weather Master 4.0"!
Next Temperature: (or -1 to quit)? -1
No temperatures were entered.
```

Problem 3 - hailstone.py

Douglas Hofstadter 獲得普立茲獎的得獎著作 *Gödel, Escher, Bach* 裡面有許多有趣的數學謎題（很多問題都可以用電腦程式來計算）

在 Chapter XII, Hofstadter 提到：選一個正整數 n ，重複以下指令直到 n 變成 1：

- 如果 n 是奇數，把 n 乘 3 再加 1
- 如果 n 是偶數，對 n 除 2

在 Hofstadter 的著作中第 401 頁，他將「若一開始選定 n 為 15」的例子呈現如下：

15	is odd, so I make $3n+1$:	46
46	is even, so I take half:	23
23	is odd, so I make $3n+1$:	70
70	is even, so I take half:	35
35	is odd, so I make $3n+1$:	106
106	is even, so I take half:	53
53	is odd, so I make $3n+1$:	160
160	is even, so I take half:	80
80	is even, so I take half:	40
40	is even, so I take half:	20
20	is even, so I take half:	10
10	is even, so I take half:	5
5	is odd, so I make $3n+1$:	16
16	is even, so I take half:	8
8	is even, so I take half:	4
4	is even, so I take half:	2
2	is even, so I take half:	1

非常有趣的是，在目前數學家測試過的所有數字裡，不管過程中 n 的值上上下下起伏了幾次，最後一定會回到 1。這個概念就好像是一顆冰雹（hailstone）不斷被風盤旋起起伏伏最後落至地表（希望沒有砸到任何東西），因此，從 n 到 1 的這個數列又被稱為 **Hailstone Sequence**

請寫出一個程式，可以讓使用者輸入任意整數，並得到 **Hailstone Sequence**。如書中的圖例，所有在抵達 1 之前的數字都會被列舉出來。舉例來說，您的程式要能完美重現下圖內所有文字與數字：

```
This program computes Hailstone sequences.  
  
Enter a number: 17  
17 is odd, so I make  $3n+1$ : 52  
52 is even, so I take half: 26  
26 is even, so I take half: 13  
13 is odd, so I make  $3n+1$ : 40  
40 is even, so I take half: 20  
20 is even, so I take half: 10  
10 is even, so I take half: 5  
5 is odd, so I make  $3n+1$ : 16  
16 is even, so I take half: 8  
8 is even, so I take half: 4  
4 is even, so I take half: 2  
2 is even, so I take half: 1  
It took 12 steps to reach 1.
```

若使用者 **recompile** 後直接輸入 1，程式要能產生 0 steps 的情況（如下圖所示）

```
This program computes Hailstone sequences.  
  
Enter a number: 1  
It took 0 steps to reach 1.
```

若同學完成，可以試試看 27 這個數字會花你們幾步，才達到 1 呢？

評分標準

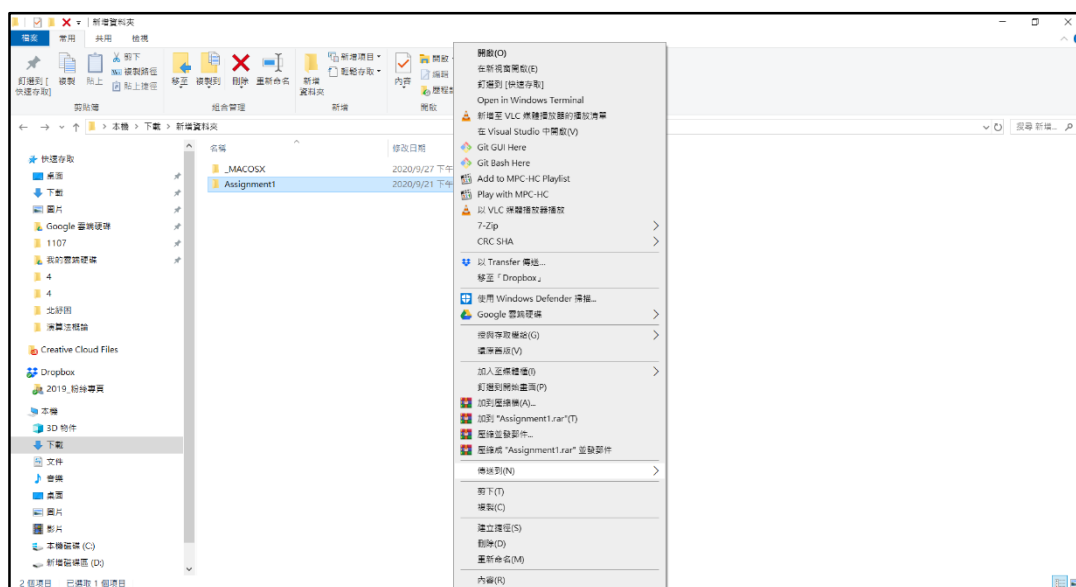
Functionality - 程式是否有通過我們的基本要求？程式必須沒有 bug、沒有 error、能順利完成指定的任務、並確保程式沒有卡在任何的無限環圈（Infinite loop）之中。這份作業每一題要測試的條件很多，尤其是 **Problem 2**，要通過各種不同真實世界的 **user inputs**。這份作業最常出錯的地方就是 **constant**，請同學在繳交前務必改變各不同的數值，測試看看是否正確

Style - 如同我們在課堂上所說，好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 **code** 去建造更多更巨大更有趣的程式。因此請大家寫**精簡扼要**的使用說明、**function** 敘述、**constant** 敘述、單行註解（簡單觀念不用說明）、**function** 名稱的命名、還有最重要的 - **Scope** 是否有對齊(Indentation)

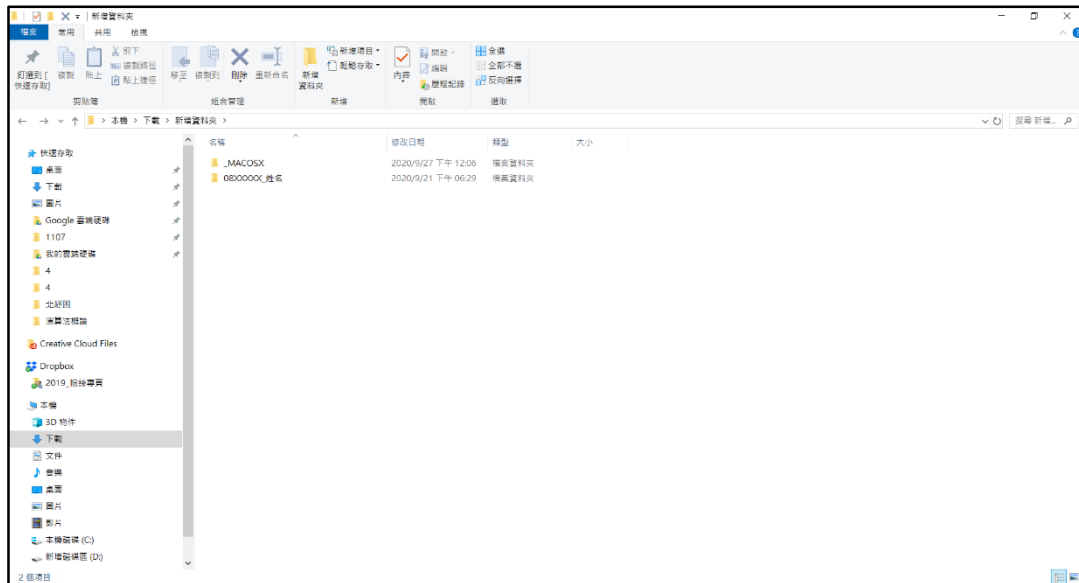
作業繳交

恭喜各位完成 **Assignment 2** 大家應該要對自己的成就感到驕傲，因為這份作業跟史丹佛大學的學生作業非常相似，代表各位成為世界最強的一群了！

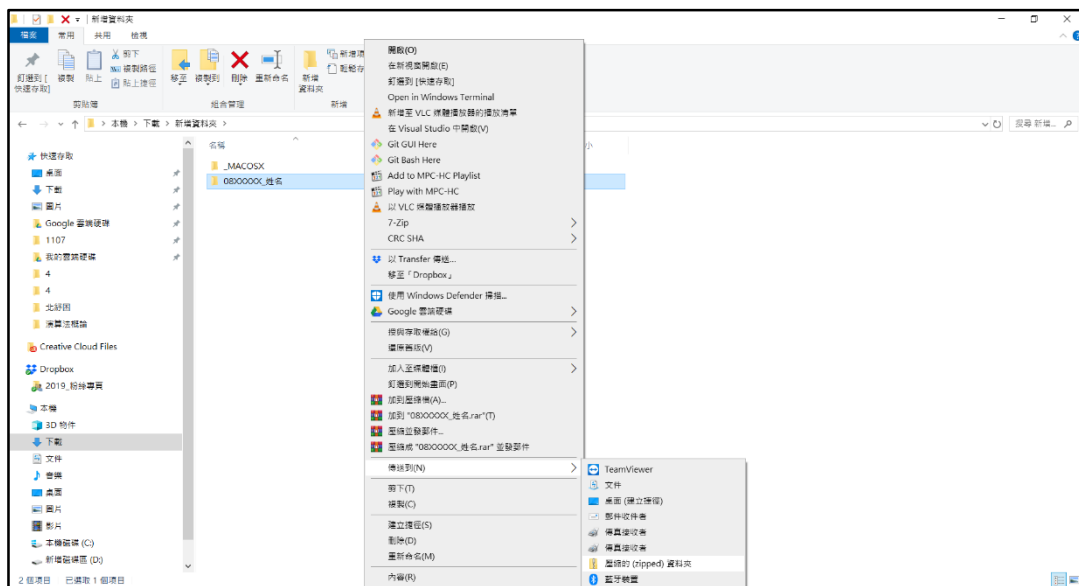
請同學於 **10/22(四) 23:59** 前依照下圖將作業上傳



找到作業資料夾，按右鍵，選擇重新命名

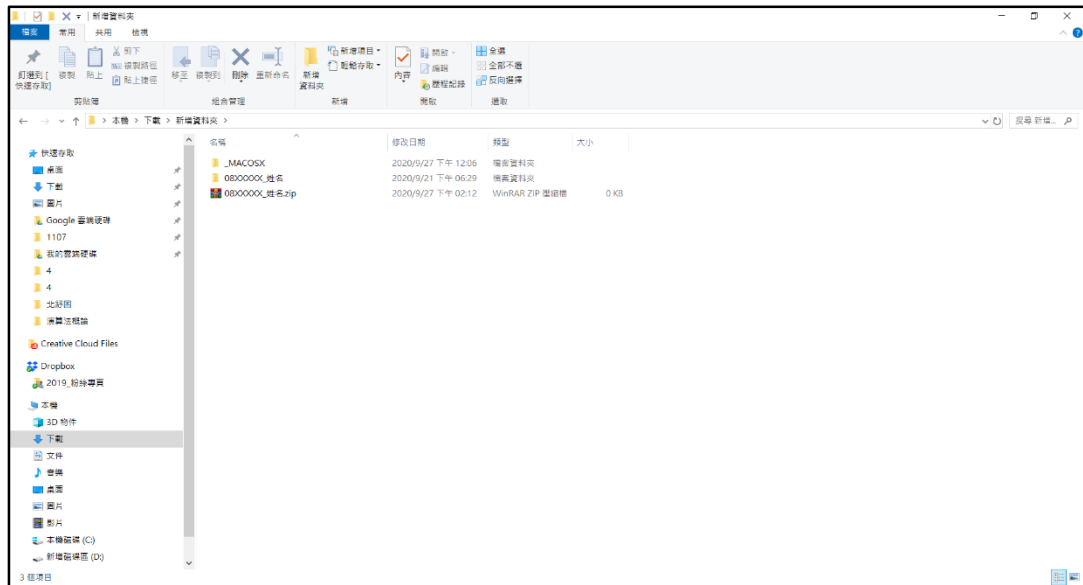


請命名成「學號_中文姓名」的格式



Windows 請點選「傳送到」->「壓縮的(zipped)資料夾」

Mac 請點選 **Compress** "學號_中文姓名"



成品如上



點選「檢視」

國立交通大學 數位教學平台

課程資訊

- 課程綱要
- 成員
- 公告列表
- 我的郵件

內容

- 大綱列表
- 同步教室 (QC3)
- 教材列表
- 作業列表
- 討論區列表
- 試卷列表

評量

- 成績

工具

- 返回我的角色

【109上】1308計算機概論 Introduction to Computer Science

Assignment 1-Test

繳交狀態

作業繳交次數	這是第1次繳交
繳交狀態	沒有繳交作業
評分狀態	尚未評分
截止日期	2020年 09月 28日(Mon) 00:00
剩餘時間	6日 5小時
最後修改	-
作業加備註	評論 (0)

 [繳交作業](#)

更改你所繳交的作業

按下「繳交作業」

國立交通大學 數位教學平台

課程資訊

- 課程綱要
- 成員
- 公告列表
- 我的郵件

內容

- 大綱列表
- 同步教室 (QC3)
- 教材列表
- 作業列表
- 討論區列表
- 試卷列表

評量

- 成績



工具

- 返回我的角色

【109上】1308計算機概論 Introduction to Computer Science

Assignment 1-Test

上傳檔案

  檔案

若要新增檔案，請將檔案拖放到這裡。

[儲存更改](#) [取消](#)

按下「檔案圖示」



選擇先前壓縮好的作業檔案

按下「上傳這一檔案」



按下「儲存更改」

The screenshot shows the National Central University Digital Learning Platform interface. The top navigation bar includes the university logo and the text '國立交通大學 數位教學平台'. The left sidebar contains a '課程資訊' (Course Information) section with links to '課程綱要' (Course Syllabus), '成員' (Members), '公告列表' (Announcement List), and '我的郵件' (My Mail). Below this is an '內容' (Content) section with links to '大綱列表' (Outline List), '同步教室 (QC3)' (Synchronous Classroom (QC3)), '教材列表' (Material List), '作業列表' (Assignment List), '討論區列表' (Discussion Forum List), and '試卷列表' (Exam Paper List). The '評量' (Assessment) section includes '成績' (Grade). The '工具' (Tools) section includes '返回我的角色' (Return to my role). The main content area displays the title '【109上】1308計算機概論 Introduction to Computer Science' and 'Assignment 1-Test'. Below this, the '繳交狀態' (Submission Status) is shown in a table:

作業繳交次數	這是第1次繳交
繳交狀態	已繳交
評分狀態	尚未評分
截止日期	2020年 09月 28日 (Mon) 00:00
剩餘時間	6 日 5 小時
最後修改	2020年 09月 21日 (Mon) 18:11
上傳檔案	08xxxxx_姓名.zip
作業加備註	評論 (0)

At the bottom right, there are two buttons: '修改已繳交的作業' (Modify submitted assignment) and '更改你所繳交的作業' (Change your submitted assignment).

出現「已繳交」確定成功



stanCode - 標準程式教育機構

Should you have any questions please feel free to contact us.