

程式與科學計算 第一次期中考（10/22 15:30~17:20，佔原始分數 11%）

**注意事項（請仔細看完再開始作答）**

- 使用系內網路時，study 的連線 ip 是 **192.168.1.23**
- 請在 study 主機上，你的家目錄下建立一個子目錄 **~/mid\_f/**。答題用的程式檔案與輸出輸入的相關檔案，請以指定的檔名全部存放這個目錄下。以程式在 study 上的執行結果評分，放在其他位置或檔名不符不予計分。
- 考試所需要的資料檔與程式範例檔都在 **/home/teachers/weitingc/hw/mid\_f/**
- 考試開放參考上課講義、筆記、範例檔、作業、作業解答、程式相關書籍、網路搜尋。
- 所有答題程式碼的撰寫、編譯、執行必須由本人獨立完成。考試期間不可與他人交談、傳遞訊息（包括網路、手機）、不可傳遞檔案，或進行其他影響考試公平性的行為。違反規定者直接取消應考資格。
- 如果不確定自己的行為是否符合規定，請事先舉手詢問監考老師或助教。
- 先完成考試作答者可以提早離開，但是要先通知助教，等助教將主機上的答題程式備份完畢後才可離場。
- 完成的程式，只要編譯後執行結果正確，就可獲得滿分。如果程式碼排版整齊，或附有清楚的註解說明，會酌予加分，最多加該題分數的 2%。

(a) 接收鍵盤輸入之參數，計算  $e$  的估計值並將結果輸出

數學常數  $e$  可由下列的無窮級數和來表示，是一種常用的泰勒級數(Taylor's series)，在數學、物理學有廣泛的應用，而此泰勒級數前  $n+1$  項的總和  $S_n$ ，可作為  $e$  的估計值。

$$e = \sum_{i=0}^{\infty} \frac{1}{i!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots \quad \text{Eq(1)}$$

$$S_n = \sum_{i=0}^n \frac{1}{i!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}, \quad n \geq 0 \quad \text{Eq(2)}$$

請寫一個 python 程式 **mida.py**，具有以下功能：

1. 接受使用者從鍵盤輸入的  $n$  值，並用四捨五入取轉換成最接近的整數。
2. 參考 Eq(2)，用迴圈累加計算出  $S_n$  之值。  
(提示：計算過程中，可使用函數 `math.factorial`)
3. 計算出的  $S_n$  數值用預設格式顯示在螢幕上（參考 b 小題的輸出結果，檢查是否計算正確）。
4. 用另一個輸出指令以預設格式在螢幕顯示  $e$  的值。

(b) 從檔案讀取參數，計算  $e$  的估計值及誤差，將結果格式化輸出並畫圖

- 輸入檔：**midb\_input.txt**，內容是數個要進行 Eq(2)計算的  $n$  值。

延續(a)，針對特定的  $n$  值，我們可以將  $S_n$  與  $e$  之間差值的絕對值定義為誤差：

$$error = |S_n - e|$$

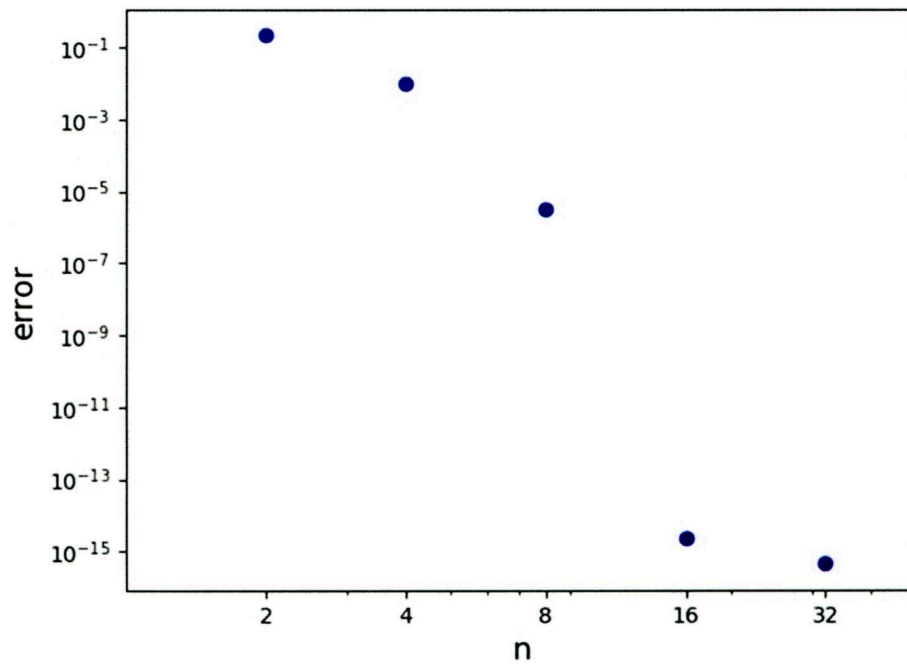
請寫一個 python 程式 **midb.py**，具有以下功能：

1. 讀取輸入檔，利用判斷式讓程式只在  $n \geq 0$  時才根據公式(2)計算  $S_n$ 。
2. 使用 python 預設的開啟與輸出指令，將  $n$ 、 $S_n$  以及  $error$  依照下列格式寫入 **midb.txt**
  - a. 第一列為 header line
  - b. 三欄資料的長度分別為 3、10、10 字元，靠右對齊，數值精確度如下

| -3- | ----10---- | ----10---- | (□為空格)

```
#nSnerror
2 2.500000 2.18e-01
4 2.708333 9.95e-03
8 2.718279 3.06e-06
16 2.718282 2.22e-15
32 2.718282 4.44e-16
```

3. 將 error 隨 n 值的變化畫出
- 資料點以藍色圓形標示，x、y 軸為對數座標
  - 座標軸刻度文字、標籤如圖示，座標軸標籤字體大小 14，x 軸範圍 1~50
  - 圖片存檔為 **midb.png** 並顯示於螢幕上

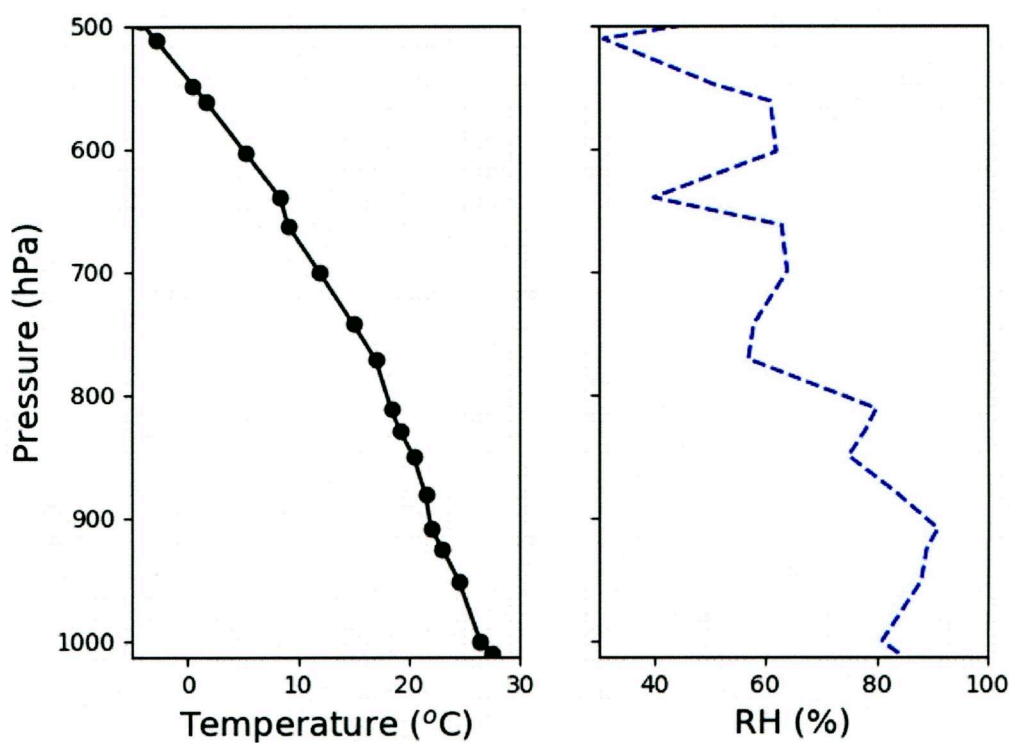


(c) 使用 `numpy` 讀入資料，畫垂直剖面

- 輸入檔：`midc_input.txt`，各直欄依序為氣壓、溫度、相對濕度

請寫一個 python 程式 `midc.py`，具有以下功能：

1. 用 `numpy` 內的函數讀取輸入檔，將各直欄分別存成一維 `numpy` 陣列 `Ps`、`Ts` 及 `RHs`。
2. 切割視窗分別畫出左右兩張圖，設定共用 y 軸
  - a. 左圖：x 軸是 `Ts`（黑色實線加圓形），範圍-5~30  
右圖：x 軸是 `RHs`（藍色虛線），範圍 30~100  
y 軸都是 `Ps`，範圍 1013~500，往上遞減
  - b. 座標軸刻度文字、標籤如圖示，標籤字體大小皆為 14
  - c. 圖片存檔為 `midc.png` 並顯示於螢幕上





(d) **numpy** 陣列運算與篩選

- 輸入檔：**mide\_input.txt** (與 c 小題相同)

低層大氣的雲量可由下面的經驗式推估：

$$CF = \left[ 30 + 185 \times \left( 1 - \frac{P}{1000} \right) \right] \times \exp \left( \frac{RH-100}{20} \right) \quad \text{Eq(3)},$$

其中 CF (%)、RH (%) 與 P (hPa) 分別為雲量、相對濕度以及壓力。

請寫一個 python 自訂函數程式 **midd\_f.py**，具有以下功能：

1. 自訂函數名稱 **cal\_cf**，接收主程式輸入氣壓與相對濕度的一維陣列，根據公式(3)計算出低層大氣雲量的一維陣列 **CF**，將 CF 回傳給主程式。自訂函數中不可使用迴圈。

請寫一個 pythony 主程式 **midd.py**，具有以下功能：

2. 引入上面所寫的自訂函數。
3. 用 numpy 內的函數讀取輸入檔，將各直欄分別存成一維 **numpy** 陣列 Ps、Ts 及 RHs。
4. 用 numpy 陣列的篩選功能，只把 P ≥ 700 的資料傳入自訂函數，將自訂函數回傳的低層大氣雲量存成一維 **numpy** 陣列 **CFs**。
5. 畫雲量的垂直剖面圖
  - a. 畫出所有 CFs (綠色點虛線) 隨 Ps 變化的垂直剖面，再加上 CFs > 20 資料點 (綠色菱形)，以及 CFs > 20 當中氣壓值最低的點 (紅色菱形)
  - b. x 軸刻度範圍 5~40，y 軸範圍 1013~700，往上遞減
  - c. 座標軸標籤如圖示，字體大小 14
  - d. 新增圖例，文字如圖示，位置在圖的右上，字體大小 12
  - e. 圖片存檔為 **midd.png** 並顯示於螢幕上

