

## 中原大學資訊工程系 演算法分析第一次機測

Deadline: 4 / 22 / 2022 (星期五)  
(限期中考當週測完，逾期不得補繳)

---

### 【程式設計說明】

1. 每組限 2~3 人，組員須固定，本學期不得任意變更。原則上以專題組員為主。
2. 組員應合作共同解題，但嚴禁跨組合作。
3. 程式設計必須使用 Python 程式語言，版本請採用目前最新版本(原則上，請直接下載與安裝 Anaconda)。
4. 可參考課本、相關書籍或網站資料等進行解題，解題方法及演算法不限，但絕對嚴禁抄襲他組程式，組員均有責任保護程式不被他組抄襲。若發現抄襲屬實，兩組均以零分計。
5. 輸入與輸出採用標準格式或讀寫檔案方式進行。
6. 每一支程式均須附上組員姓名及學號，例如：

```
# 演算法分析機測  
# 學號: 10827XXX / 10827XXX  
# 姓名: 江○○ / 李○○  
# 中原大學資訊工程系
```

程式命名依該組學號在前之同學 [學號+題號] 為原則。例如：

```
10827001_1.py  
10827001_2.py
```

---

### 【機測須知】

1. 評分以解題成功之題數多寡與執行時間決定。
2. 程式必須能處理不同之輸入資料(但輸入格式與範例相同)，並輸出正確結果(輸出格式必須與範例相同)，組員應能說明程式設計內容，方可視為成功。程式之輸出結果錯誤、輸出格式與範例不符、或在執行後超過 60 秒仍未結束，均視為失敗。若程式測試失敗給予基本分數，未繳交程式則以零分計。
3. 本機測於規定之期限前，各組應攜帶程式原始碼至電學大樓 603 室找助教測試(電話：265-4726)，每組限繳交一次，不可分題或多版本繳交，逾期不得補繳。
4. 助教將使用不同之輸入資料作為測試與評分依據，同學應在繳交前充分測試程式。
5. 機測成績納入學期平時成績計算，請同學把握！

---

指導教授: 張元翔

## I. 神秘的門 (Secret Door)

張小明率領的尋寶探險隊，在尋寶過程中遇到一個神秘的門。這個神秘的門無法使用其他方法打開，必須先解開特定的謎題。



神秘的門上有許多石盤，每個石盤上有一個英文單字。這些石盤可以被重新排列，解謎的關鍵是：「石盤上的每個英文單字的第一個字母，必須與前一個石盤上的英文單字的最後一個字母相同。」例如：apple 後面可以接 egg，egg 後面可以接 grape 等。

請您設計程式，協助張小明的尋寶探險隊，打開這個神秘的門。

【註】本題允許使用暴力法解題。

### 輸入說明：

第一列為正整數  $n$  ( $3 \leq n \leq 9$ )，代表石盤的數量，每個石盤上有一個英文單字。接著，每一列為石盤上的英文單字，且均為小寫字母。 $n = 0$  代表結束。

### 輸出說明：

如果可以安排石盤的順序，使得每個單字的第一個字母與前一個石盤的單字最後一個字母相同，則輸出「Can be opened.」；否則，輸出「Can not be opened.」。若可開啟，則輸出排列的順序。

### 輸入範例：

```
3
apple
egg
grape
5
bacon
beef
fish
hamburger
rib
3
dog
```

cat

pig

0

**輸出範例:**

Secret Door 1

Can be opened.

apple-egg-grape

Secret Door 2

Can be opened.

beef-fish-hamburger-rib-bacon

Secret Door 3

Can not be opened.

## II. 最大子陣列問題 (Maximum-Subarray Problem)

**最大子陣列問題** (Maximum-Subarray Problem) 在電腦演算法中是一個相當重要的問題，描述如下：

給定一整數陣列 (Array)，其中可能包含正或負整數，目的是找到子陣列 (Subarray)，即連續元素整數和，且其總和最大。請參考課本 (講義) 之 Divide-and-Conquer 演算法，設計程式解最大子陣列問題。

### 輸入說明：

輸入包含幾組資料，每組資料以一正整數  $n$  開頭，代表輸入之整數個數，若為 0 則代表結束。接著為  $n$  個整數，每個整數以空格隔開。你可以假設所有整數均介於  $-100 \sim 100$  之間。

### 輸出說明：

根據每組資料輸出最大子陣列的最小索引、最大索引及最大總和(假設索引值從 1 開始)。

### 輸入範例：

```
8
-2 1 -3 4 -1 2 1 -5
16
13 -3 -25 20 -3 -16 -23 18 20 -7 12 -5 -22 15 -4 7
0
```

### 輸出範例：

```
Low = 4, High = 7, Sum = 6
Low = 8, High = 11, Sum = 43
```

### III. 渡河問題 (River Crossing Problem)

假設有一主人有  $M$  隻狼與  $N$  隻羊 ( $M \leq N$ ) 需從西岸 (W) 渡河到東岸 (E)，其使用的船不含主人本身需乘載至少一隻才渡河，最多只能乘載兩隻。由於狼會吃羊的緣故，為了安全的考量，每次渡河時，不管在船上、西岸、或東岸，羊的數目均不能比狼少，除非羊的數目等於零。

若定義  $(X, Y, P)$  來表示渡河時的不同狀態，其中  $X, Y$  分別代表狼、羊在西岸的數目， $P$  代表渡河前船的位置 (W or E)。舉例而言，最初的狀態是  $(M, N, W)$ ，最後的狀態是  $(0, 0, E)$ ，其中一安全的狀態為  $(m, n, P)$ ，則  $m \leq n$  (除非  $n=0$ )，且  $M-m \leq N-n$  (除非  $N-n=0$ )，亦即不管在船上、西岸或東岸，羊的數目均不能比狼少。

試寫一程式解決渡河問題。現以下為範例，假設剛開始時，狼與羊各有兩隻 ( $M = 2, N = 2$ )，則其渡河的次序為 (在渡河次數最少的情況下)：

$(2, 2, W) \rightarrow (0, 2, E) \rightarrow (1, 2, W) \rightarrow (1, 0, E) \rightarrow (2, 0, W) \rightarrow (0, 0, E)$

其中渡河的次序中不應有重複。

#### 輸入說明：

每列代表輸入初始狼與羊的數目  $M$ 、 $N$  (均為正整數，且  $M \leq N$ )，0 0 表示結束。

#### 輸出說明：

顯示渡河的次序。

#### 輸入範例：

2 2

0 0

#### 輸出範例：

2, 2, W

0, 2, E

1, 2, W

1, 0, E

2, 0, W

0, 0, E

#### IV. 撲克牌 24 點遊戲 (Poker 24-Point Game)

現有一副撲克牌，每張牌的數值均介於 1~13 之間（即 A 視為 1，J、Q、K 分別視為 11、12、13，本問題不考慮花色）。撲克牌 24 點遊戲的玩法如下：

- 從這副撲克牌中任意抽出 4 張牌給玩家（數值有可能重複）。
- 玩家可將這 4 張牌的數值任意排列，然後採用加、減、乘、除四則運算（即+、-、\*、/等），運算時允許小數存在，並且可以使用括號。但是，每張牌只可以使用一次。
- 玩家嘗試建構一運算式，使其運算結果為 24。

請您根據上述遊戲規則，試寫一程式幫忙玩家解決問題。

##### 輸入說明：

每列表示任意抽出的一組撲克牌，分別為 4 張牌的數值（中間以空格隔開），0 0 0 0 則代表結束。

##### 輸出說明：

若能得到運算結果為 24，則輸出相對應的運算式，否則輸出無解 (No Solution)。

##### 輸入範例：

```
11 8 3 5
3 3 7 7
9 9 6 2
1 1 1 2
0 0 0 0
```

##### 輸出範例：

```
(11-8)*(3+5)=24
7*(3+3/7)=24
9*(2+6/9)=24
No Solution
```

**注意：**運算應依先乘除後加減， $7*(3+3/7)$  若輸出為  $7*(3+(3/7))$  仍視為是不正確的輸出。

## V. 拼圖 (Puzzle)

拼圖 (Puzzle) 遊戲是常見的益智遊戲，典型的拼圖如下：



海賊王(One Piece) 原圖



海賊王(One Piece) 拼圖

試設計程式讀取拼圖之影像檔，例如：One\_Piece1.bmp，輸出拚好的影像檔，例如：One\_Piece1\_result.bmp。

【註】本問題的時間限制為 5 分鐘。

拼圖的詳細說明如下：

- 每張影像均為彩色影像，影像大小為  $1920 \times 1080$  像素(寬 $\times$ 高)。
- 每張小拼圖的大小為  $120 \times 120$  像素，因此總共有  $16 \times 9$  塊小拼圖。
- 每張小拼圖僅在幾何空間中進行任意平移，並無套用旋轉或其他操作。任意平移後進行拼接，次序以隨機方式進行。

#### 輸入說明：

採讀檔方式進行，影像檔案格式採用.bmp。

#### 輸出說明：

輸出拚好的影像檔，影像檔案格式採用.bmp。

#### 輸入範例：

請輸入影像檔: One\_Piece1.bmp [Enter]

#### 輸出範例：

輸出影像檔: One\_Piece1\_result.bmp

---

相關技術說明如下：

- 安裝 OpenCV 進行數位影像檔的讀寫。

```
pip install opencv-python
```

- 典型的讀檔程式範例如下：

```
import numpy as np
import cv2

img = cv2.imread( "One_Piece1.bmp", -1 )
cv2.imshow( "Example", img )
cv2.waitKey( 0 )
```

- 彩色影像是以 3 維陣列存取，即 `np.array( [1080, 1920, 3], dtype = "uint8" )`，資料型態為 `uint8` (每個像素包含 3 個 Bytes，分別表示 B、G、R 三原色，又稱為通道。每個通道的色彩值是以 1 個 Byte 表示，介於 0~255 之間)。
- 本問題無法使用暴力法求解，請您思考比較有效的演算法。