

109 計算機程式設計 (Computer Programming 1279)

bhw06 說明：

1. Deadline: 2020/12/23 23:59 (不接受遲交)
2. llearning 上傳標題：學號_bhw06
3. 程式碼檔案名稱(檔名錯誤不計分)：
 - 第 1 題：學號_p1.c
 - 第 2 題：學號_p2.c
 - 第 3 題：學號_p3.c
 - 第 4 題：學號_p4.c
4. 程式內開頭第 1 行到第 4 行，以註解的方式寫下列資訊：
/*
* 系級：
* 學號：
* 姓名：
* 聯絡電子郵件：
*/
5. 每筆測資限制執行時間為 8 秒

bhw06 題目：

1. 在線性代數中，行列式 ([Determinant](#)) 是一個很有用的值，可以從方陣的元素中計算得出。行列式也具有許多有用的特性。例如：假設我們有一個方陣 A。那麼當且僅當 A 是不可逆時，A 的行列式等於零。

現在給定 N×N 的方陣 A，請計算 A 的行列式。(25%)

提示：

- (1). 我們可以定義 2×2 矩陣的行列式如下：

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

- (2). 我們可以定義 3×3 矩陣的行列式如下：

$$\begin{aligned} |A| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= aei + bfg + cdh - ceg - bdi - afh. \end{aligned}$$

- (3). [Laplace expansion](#)

輸入輸出規範：

- $1 \leq N \leq 8$
- $-16 \leq \text{矩陣元素大小} \leq 15$
- 行列式值大小可能會超過 int 的上限
- 輸入之方陣中的元素以一個空白間隔
- 輸入格式：
N
N×N 方陣

程式執行範例 1：

```
2
1 0
0 1
1
```

程式執行範例 2：

```
3
1 2 3
4 5 6
7 8 9
0
```

程式執行範例 3：

6
-8 -3 14 9 -4 10
15 -14 -1 7 13 -9
-2 13 7 13 9 -13
-7 -1 -11 -1 3 5
7 -16 2 -6 3 10
6 8 15 -4 -14 -10
-2687875

程式執行範例 4：

5
1 13 -1 -11 -8
-7 -3 6 -3 -7
9 13 13 8 -3
6 -9 4 -7 8
-15 -12 14 -9 -4
-400040

2. 在某年冬天的紐西蘭電力危機期間（由於降雨不足，從而導致水壩水位低下），制定了應急計劃，以系統且完全公平的方式關閉該國的電力。該國分為 N 個地區（奧克蘭是 1 號地區，惠靈頓是 13 號地區）。將“隨機”選擇一個數字 m ，首先將在區域 1（顯然是最公平的起點）中關閉電源，接下來是在該區域之後的第 m 個區域將電力關閉，若超過 N 之後會從 1 重新開始繼續數，且忽略已經關閉的區域。例如，如果 $N = 17$ 且 $m = 5$ ，則將按以下順序關閉區域的電源：1、6、11、16、5、12、2、9、17、10、4、15、14、3、8、13、7。

本題問題在於，最後關閉惠靈頓顯然是最公平的做法（畢竟，這是電力總部所在的位置），因此對於給定的 N ，需要仔細選擇“隨機”數 m ，以便使得最後一個選擇的區域是區域 13。

請設計一個程式，該程式將讀取區域數 N ，然後確定最小數 m ，以確保惠靈頓（第 13 區）可以在全國其他地區都無法使用的情況下正常運行。

(25%)

輸入輸出規範：

■ $13 \leq N \leq 100$

程式執行範例 1：

17
7

程式執行範例 2：

18
17

程式執行範例 3：

19
11

3. 輸入 $X+Y$ 與 $X-Y$ 的值，求 X, Y 分別為多少。(25%)

輸入限制：

- 輸入皆為整數
- $-100 \leq \text{輸入數字} \leq 100$
- 保證有唯一解，且 $X、Y$ 皆為整數

程式執行範例 1：

```
3 1  
2 1
```

程式執行範例 2：

```
2 -2  
0 2
```

4. Ta 有一顆骰子，由於他想知道這顆骰子是否公平，於是就請了 LAB 養的猴子們幫忙擲骰子，並把點數記錄在一些 csv 檔案裡。然而在做紀錄的時候猴子有時候會犯錯，根據 Ta 從前的觀察，猴子登記錯點數（點數不是 1~6）的機率是 $\frac{1}{1000}$ （如下圖(一)），不小心跳過某個儲存格（留下空白）的機率是 $\frac{1}{20000}$ （如下圖(二)）。現在 Ta 想麻煩同學，幫忙統計一下所有 csv 檔案裡各個點數的出現次數、猴子登記錯點數的次數、跳過儲存格的次數和個別的比率，來確定骰子是否公平和猴子是否有進步，請把統計結果寫到 result.csv 和程式碼一起上傳到 llearning。(25%)

備註：

- 所有 csv 檔放在 diceData 資料夾內，ZIP 檔要記得解壓縮。
- 共有 diceData000000.csv ~ diceData000999.csv，計 1000 個檔案，命名方式皆相同 diceData + 編號 6 位數 + .csv。
- 每個檔案裡面都有 100,000 筆資料，每行都是 100 筆資料。
- 附檔 result1.csv、result30.csv、result50.csv 分別是前 1、30、50 個檔案的統計結果，可以參考統計完的檔案長成怎樣，也可以拿來比對自己做的結果。
- 不知道檔案怎麼讀可以用記事本開檔案，看一下檔案的格式。
- 可能會使用到 sprintf 函式來弄出每個檔案的檔名。
- 這題的工作沒有任何的輸入或輸出也沒有執行時間限制，只要上傳程式碼跟最終結果的 result.csv。

26	1	1
2	6	3
6	1	2
4	3	26

圖(一) 猴子輸入錯誤

6	5	5
3		4
5	5	6
1	2	5

圖(二) 猴子跳過儲存格(圖片錯誤所在處：diceData000000.csv row 667, column BK)