

# 110 計算機程式設計 (Computer Programming 1279)

上機作業、考試繳交方式：

每次上課都會有上機作業，必須在課堂上完成，有 1~2 小時可用，在下課前上傳到 I-learning，上傳時只需要上傳 **.c 檔**。除非有事先請假，否則不接受課後補交！

上機作業與考試的格式：

1. 依照下列格式命名：學號\_p#.c (# => 題號)

例：

**作業 01 第一題** => 4109056099\_p1.c

**加分題 01 第一題** => 4109056099\_b1.c

由於 ilearning 評分上限為 100 分，為能順利評筆，請同學依照以下方式上傳作業：

將作業題上傳到[hw\_xx 課堂作業(基本題)]項目。

將加分題上傳到[hw\_xx 課堂作業(加分題)]項目。

2 上傳後可以上傳修改版本(在下課以前)，如果想要上傳更新的版本，檔案名稱維持一樣直接上傳，助教只會對最新的版本評分。

3. 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：

\*/

4. 程式輸入輸出部分有底線的為 input 沒有的 output

例：100 (100 為輸入)

212.00 (212.00 為輸出)

5. 請注意 不要使用下列兩個 function 用了作業會錯

System(“pause”)

fflush()

6. **每筆測資限制執行時間為 8 秒**

### Hw10 基本題：

1. 請根據之前的氣泡排序法做修改，並以指標的方式實作，而非使用索引的方式。試著寫一個函式 `int *bubble_sort(int*, int)` 來排序輸入的陣列，接著把排序好的陣列的起始位址回傳至 `main` 並輸出。函式參數依序代表陣列起始位置、陣列的長度，回傳值表示陣列的起始位置。

題目規範：

- $1 \leq N \leq 20$ 。
- 輸入格式：  
    N  
    N 個數字
- 每個數 `int` 皆可表示的範圍

程式輸出範例 1：

```
10
9 5 1 7 2 4 8 3 0 6
0 1 2 3 4 5 6 7 8 9
```

程式輸出範例 2：

```
7
9 18 22 7 56 4 99
4 7 9 18 22 56 99
```

程式輸出範例 3：

```
15
6 77 80 26 128 66 94 -10 72 1 37 69 44 25 -77
-77 -10 1 6 25 26 37 44 66 69 72 77 80 94 128
```

2. 輸入一連串的字元，請將這串字元進行順時鐘旋轉，並輸出所有的結果。  
(25%)

題目規範：

- $1 \leq \text{總字元長度} \leq 50$
- 字元內容僅含大小寫字母

程式輸出範例 1：

Jinkela  
inkelaJ  
nkelajI  
kelaJin  
elajink  
laJinke  
aJinkel  
Jinkela

程式輸出範例 2：

Jason  
asonJ  
sonJa  
onJas  
nJaso  
Jason

3. 請根據之前的矩陣乘法做修改，並以指標的方式實作，而非使用索引的方式。試著寫一個函式 `void mul(int(* A)[10], int(* B)[10], int(*ret)[10])` 來計算兩矩陣相乘，接著把結果存在 `ret` 矩陣裡並在 `main` 印出結果。函式參數依序代表 `A` 陣列起始位置、`B` 陣列起始位置、`ret` 陣列起始位置。

題目規範：

- 請使用全域變數宣告 `m, n, p`， $1 \leq m, n, p \leq 10$ ，並在 `main` 裡賦予值。
- 在 `main` 裡宣告矩陣 `A[10][10]`、`B[10][10]`、`ret[10][10]`。
- 根據 `m, n` 賦予矩陣 `A` 的值、根據 `n, p` 賦予矩陣 `B` 的值。
- 矩陣 `A`、`B` 賦予的值範圍為  $-100 \sim 100$ 。

假設  $m = 2$ ， $n = 2$  則需要在 `A` 矩陣裡輸入  $2 \times 2$  個值。

假設  $n = 2$ ， $p = 4$  則需要在 `B` 矩陣裡輸入  $2 \times 4$  個值。

程式輸出範例 1：

3 2 3

1 4

2 5

3 6

7 8 9

10 11 12

47 52 57

64 71 78

81 90 99

程式輸出範例 2：

3 3 5

7 5 12

4 14 9

33 10 1

8 9 14 22 7

7 21 76 89 44

3 41 20 15 23

127 660 718 779 545  
157 699 1300 1469 851  
337 548 1242 1631 694

程式輸出範例 3：

3 5 3

-10 5 2 37 5  
21 51 99 52 15  
10 -8 -75 42 8

-12 -8 -7  
65 10 55  
91 4 67  
32 77 54  
64 25 8

2131 3112 2517  
14696 5117 12219  
-5609 2974 -3203

4. 輸入一個長度為  $N$  的數列： $A_1, A_2, A_3, \dots, A_n$ ，請依序輸出以下  $N$  個問題的答案：

從數列中移除  $A_1$ ，其他數字中的最大值為？

從數列中移除  $A_2$ ，其他數字中的最大值為？

從數列中移除  $A_3$ ，其他數字中的最大值為？

...

從數列中移除  $A_n$ ，其他數字中的最大值為？(25%)

輸入格式：

$N$

$A_1$

$A_2$

...

$A_n$

輸入輸出規範：

■  $2 \leq N \leq 200,000$

■  $1 \leq A_i \leq 200,000$

■ 所有輸入數字皆為整數

程式執行範例 1：

3

1

4

3

4

3

4

程式執行範例 2：

2

5

5

5

5

Hint:

考慮最大值不重複的情況下，

如果移除的不是數列中的最大值，那答案是？

如果移除的是數列中的最大值，那答案是？

->找最大的兩個值

再考慮如果最大值有重複出現的話，可不可以用一樣的方式求出答案。

或有一種方法不需要找出最大的兩個值。

### Hw10 加分題：

1. 輸入  $N$  不同的正整數，請輸出所有在這  $N$  個數字中取  $M$  個的組合可能，請依照字典序(lexicographic order)順序輸出。(25%)

題目規範：

- $1 \leq M \leq N \leq 10$
- 輸出數字之間以一個空白間隔
- 輸入格式：  
     $N$   $M$   
     $N$  個數字

程式輸入輸出範例 1：

```
53  
13579  
135  
137  
139  
157  
159  
179  
357  
359  
379  
579
```

程式輸入輸出範例 2：

```
53  
97153  
135  
137  
139  
157  
159  
179  
357  
359  
379  
579
```

2. 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](#)

題目規範：

- $1 \leq N \leq 8$
- $-16 \leq \text{矩陣元素大小} \leq 15$
- 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