

110 計算機程式設計

(Computer Programming 1279)

109.11.29

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，否則不接受事後補交！

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：

基本題 第一題 => 4109056099_**p1**.c

加分題 第一題 => 4109056099_**b1**.c

2. 上傳後可以上傳修改版本(在遲交期限以前)，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

hw09 基本題：

1. 輸入兩個數字 N, M，寫一個函式 `void function_name(int *, int *)`，將讀取輸入的兩個變數數值交換後，於 `main` 中輸出。

程式輸入輸出範例 1：

1 100
100 1

程式輸入輸出範例 2：

2 3
3 2

程式輸入輸出範例 3：

-8 9
9 -8

2. 輸入一個分數 N/M ，寫一個函式 `void function_name(int, int, int*, int*)` 將分數化為最簡分數後，將其結果存入後兩個指標所指之記憶體位址內，並於 `main` 中輸出結果。函式傳入的參數依序代表分子、分母、分子約分後存入的記憶體位址、分母約分後存入的記憶體位址。

程式輸入輸出範例 1：

10/2

5/1

程式輸入輸出範例 2：

1284/651

428/217

程式輸入輸出範例 3：

8/20

2/5

3. 輸入一串數字(數字不會重複)，將其存入一個陣列，接著請寫一個函式 `int* function_name(int*, int)` 找到最大值所在的記憶體位址並回傳至 `main`，並於 `main` 中輸出最大值為多少。函式參數依序代表陣列起始位址、輸入數字的數量，回傳值表示最大值所在的位址。

輸入格式：

`N`

`num1 num2 num3 ... numN`

第一行的 `N` 表示會輸入幾個數字

第二行為輸入的 `N` 的數字

$$1 \leq N \leq 100$$

程式輸入輸出範例 1：

5

1 2 8 4 -5

8

程式輸入輸出範例 2：

3

-8 10 15

15

4. 輸入一串數字，將其存入一個陣列，並寫一個函式 `void function(int*, int*, int)`，將輸入數字反序存入另一個陣列，並將結果於 `main` 中輸出。函式參數依序代表原本儲存數字的陣列起始位址、要將數字反序存入的陣列起始位址、輸入數字的數量。

輸入格式：

N

$num_1 \ num_2 \ num_3 \ ... \ num_N$

第一行的 N 表示會輸入幾個數字

第二行為輸入的 N 的數字

$$1 \leq N \leq 100$$

程式輸入輸出範例 1:

4

1 8 -10 2

2 -10 8 1

程式輸入輸出範例 2:

5

1 2 10 -10 2

2 -10 10 2 1

hw09 加分題：

1. 一有 n 個整數的序列我們稱為 jolly jumper，如果相鄰的 2 個數其差的絕對值恰好為 1 到 $n-1$ 。例如：

1 4 2 3

就是 jolly jumper ($n=4$)。因為相鄰 2 數的差的絕對值為 3,2,1，就是 1 到 $n-1$

但是

1 4 2 -1 6

不是 jolly jumper ($n=5$)。因為相鄰 2 數的差的絕對值為 3,2,3,7，並非 1 到 $n-1$

寫一個程式來判斷一個整數序列是否為 jolly jumper。

輸入說明：

第一個正整數為 n ($n < 3000$)，代表此整數序列的長度。接下來有 n 個整數，代表此整數序列。

輸入限制：

■ $n < 3000$

程式輸入輸出範例 1：

4

1 4 2 3

Jolly

程式輸入輸出範例 2：

5

1 4 2 -1 6

Not jolly

2. 輸入 N 個整數 $num_1, num_2, \dots, num_N$,
對滿足 $1 \leq i < j \leq N$ 的數對 (i, j) 計算 $num_i * num_j$ 的總和 $\% (10^9 + 7)$,
也就是計算 $[\sum_{i=1}^{N-1} \sum_{j=i+1}^N (num_i * num_j)] \% (10^9 + 7)$

輸入限制：

- $2 \leq N \leq 2 * 10^5$
- $0 \leq num_i \leq 10^9$

Hint:

$$(A + B) \% P = [(A \% P) + (B \% P)] \% P$$

$$(A * B) \% P = [(A \% P) * (B \% P)] \% P$$

注意所有計算中可能產生的 **overflow** 問題，並注意資料型態。
思考簡化求和方式，以避免超過執行時限。

程式輸入輸出範例 1：

```
3
1 2 3
11
```

說明：符合 $1 \leq i < j \leq N$ 的數對 (i, j) 有 $(1,2) (1,3) (2,3)$ ，因此答案為

$$(num_1 * num_2 + num_1 * num_3 + num_2 * num_3) \% (10^9 + 7)$$
$$= (1 * 2 + 1 * 3 + 2 * 3) \% (10^9 + 7) = 11$$

程式輸入輸出範例 2：

```
4
141421356 17320508 22360679 244949
437235829
```

範例大測試資料請參考附件 txt 檔

b2_ex1.txt

ans: 995093147

b2_ex2.txt

ans: 524458326

如何測試：

將編譯完成之執行檔 (XXX.exe) 與測試資料 (XXXX.txt) 置於同一個目錄下

開啟 cmd

輸入 XXX.exe < XXXX.txt