

本地練習題

5 題 · 本地 gcc/clang · 總分 動手寫

做法：讀題 → 自己寫 → `./run.sh 你的檔.c` 編譯執行 → 對答案看 [參考解答/](#)。用 `scanf` 的程式執行後要自己輸入。

參考解答是可編譯執行的程式檔（[參考解答/01_max3.c](#) … [05_swap.c](#)）。注意：函式回傳型態要對、遞迴一定要有基底條件、傳值改不到呼叫端。

§ 1 練習題

練習 1：三數最大（定義／呼叫／回傳）

寫一個函式 `int max3(int a, int b, int c)`，回傳三個數中最大的。main 讀入三個整數、呼叫它、印出結果。

引導步驟

1. 函式內先假設第一個最大，再和其餘比、留大的。
2. main 用 `scanf("%d %d %d", &a, &b, &c)` 讀三個數。

易錯

- 函式忘了 `return`；回傳型態寫成 `void` 卻又要值。 **參考解答：** [參考解答/01_max3.c](#)

練習 2：判斷質數（函式封裝邏輯）

寫一個函式 `int is_prime(int n)`，是質數回傳 1、不是回傳 0。main 讀入 `n`，依結果印 `prime` 或 `not prime`。

引導步驟

1. 小於 2 不是質數；用迴圈試除 2 到 `i*i <= n`，整除就回傳 0。
2. main 用 `is_prime(n) ? "prime" : "not prime"`。

易錯

- 把判斷邏輯全塞 main、不寫函式；迴圈條件 `i <= n` 太慢但仍對，`i*i <= n` 較佳。 **參考解答：** [參考解答/02_is_prime.c](#)

練習 3：遞迴階乘

用遞迴寫 `fact(n)` 算 `n!`（不准用迴圈）。main 讀入 `n` 印出結果。

引導步驟

1. 基底條件：`n <= 1` 時回傳 1。
2. 遞迴步驟：`return n * fact(n - 1)`，每次 `n` 變小、終會碰到基底。
3. 大數會溢位，回傳型態用 `long long`、印 `%lld`。

易錯

- 忘了基底條件 → 無窮遞迴 → Stack Overflow。 **參考解答：** [參考解答/03_fact_recursive.c](#)

練習 4：遞迴次方

用遞迴寫 `power(base, exp)` 算 `base` 的 `exp` 次方 (`exp >= 0`)。main 讀入 `base exp` 印出結果。

引導步驟

1. 基底條件：`exp == 0` 時回傳 1 (任何數的 0 次方是 1)。
2. 遞迴步驟：`return base * power(base, exp - 1)`。

易錯

- 基底條件寫成 `exp == 1` (漏掉 0 次方)；遞迴步驟忘了讓 `exp` 變小。 **參考解答：**
[參考解答/04_power_recursive.c](#)

練習 5 (重要)：交換兩變數 (傳值 vs 傳位址)

寫 `void swap(int *a, int *b)` 交換兩個變數的值。main 讀入 `x y`、呼叫 `swap(&x, &y)`、印出交換後的 `x y`。想想：若參數改成 `int a, int b` (傳值) 為什麼換不成？

引導步驟

1. 傳值只拿到複本，改複本動不到呼叫端 → 必須傳**位址**。
2. 函式收 `int *a, int *b`；用暫存 `int t = *a; *a = *b; *b = t;`。
3. 呼叫時傳位址：`swap(&x, &y)`。(指標細節見 [[指標與記憶體]])

易錯

- 用傳值版 `swap(int a, int b)` (換的是複本，呼叫端不變)；呼叫忘了加 `&`。 **參考解答：**
[參考解答/05_swap.c](#)