

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_sizeof.c](#) … [05_celsius.c](#)）。永遠初始化變數；浮點別用 `==` 比較。

§ 1 練習題

練習 1：用 `sizeof` 探查型態大小

寫程式印出 `int`、`float`、`double`、`char` 各佔幾個 byte。

引導步驟

1. `sizeof(int)` 回傳位元組數，型別是 `size_t`，用 `%zu` 印。

易錯

- 用 `%d` 印 `sizeof`（型別不符）。 **參考解答：** [參考解答/01_sizeof.c](#)

練習 2：char 與 ASCII

讀入一個小寫英文字母，印出它的 ASCII 值，並轉成大寫印出。

引導步驟

1. `char c; scanf(" %c", &c);` (`%c` 前留空白跳過換行)。
2. `%d` 印 ASCII；大寫 = `c - 32` (`'a'→'A'` 差 32)。

易錯

- 忘了 `%c` 前的空白；以為 `'0'` 等於 `0`。 **參考解答：** [參考解答/02_char_case.c](#)

練習 3：觀察整數溢位

印出 `int` 的最大值 `2147483647`，再印出「最大值 + 1」的結果，觀察溢位繞回。

引導步驟

1. `int x = 2147483647;`，印 `x` 與 `x + 1`。
2. 想想 32 bits、1 bit 正負號，+1 會繞到哪。

易錯

- 以為會 crash（不會，是繞回最小負數，屬未定義行為）。 **參考解答：** [參考解答/03_overflow.c](#)

練習 4：浮點精度

印出 $0.1 + 0.2$ 到小數 17 位，並分別用 `==` 與「容差」判斷它是否等於 0.3 。

引導步驟

1. `double a = 0.1 + 0.2; , printf("%.17f\n", a);` 。
2. `if (a == 0.3)` 與 `if (fabs(a - 0.3) < 1e-9)` (需 `#include <math.h>`) 。

易錯

- 以為 `0.1+0.2 == 0.3` 成立；浮點要用容差比。 **參考解答：** [參考解答/04_float_precision.c](#)

練習 5：攝氏轉華氏

讀入攝氏溫度 (`double`)，印出華氏到小數一位 ($F = C \times 1.8 + 32$)。

引導步驟

1. `double c; scanf("%lf", &c);` (`double` 用 `%lf` 讀) 。
2. `%.1f` 印 `c * 1.8 + 32` 。

易錯

- 用 `int` 存溫度 (小數不見)；`scanf` 讀 `double` 用了 `%f` (要 `%lf`)。 **參考解答：** [參考解答/05_celsius.c](#)