整章測驗卷（答案卷）

時限 70 分鐘 · 總分 100

本卷為**答案與評分要點**。每題列正解、關鍵步驟與常見扣分點（檢誤）。位址以 `sizeof(int)=4` 計。

§ 1　解答

Q1 解答（預測輸出｜15 分）

解答

輸出 `10 20`。逐步：

1. 初始 `a=1, b=2`，`p` 指向 `a`。
2. `*p = 10` 改 `p` 指向的 `a` → `a = 10`。
3. `p = &b` 後 `p` 改指 `b`；`*p = 20` → `b = 20`。

最終 `a = 10`、`b = 20`。記憶體圖：`p` 的箭頭先指 `a`、後改指 `b`。

檢誤

- 誤以為 `p = &b` 後 `a` 也被改（`a` 在第 2 步後就固定為 10）。
- 把 `*p = 10` 當成只改 `p` 不改 `a`。

Q2 解答（指標算術｜15 分）

解答

`p = &arr[1]`。

- (a) `*(p+2) = arr[3] = 8`
- (b) `p[3] = *(p+3) = arr[4] = 10`
- (c) `p` 的位址 = `0x300 + 1×4 = 0x304`；`p+2 = 0x304 + 2×4 = 0x30c`

檢誤

- 把 `*(p+2)` 當成 `arr[2]`（忘了 `p` 從 `arr[1]` 起算）。
- 位址只加 2 而非 `2×sizeof(int)`。

Q3 解答（手寫函式｜20 分）

解答

```c
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
/* 呼叫：swap(&x, &y); */
```

C 是傳值，傳值版 `void swap(int a, int b)` 只交換函式內的複本，呼叫端的 `x`、`y` 不變；要改呼叫端就得把位址傳進去，函式透過 `*` 寫回。

檢誤

- 參數寫成 `int a, int b`（傳值，交換無效）。
- 函式內漏 `*`，直接交換指標而非指向的值。
- 沒說明傳值為何無效（占說明分）。

Q4 解答（抓 bug｜15 分）

解答

bug：懸空指標（回傳區域變數位址）。`local` 在 stack，函式返回時其空間即回收，回傳的位址失效，呼叫端讀寫是未定義行為。兩種修法：

```
/* 修法 1：改放 heap，呼叫端負責 free */
int *make_array(void) {
    int *a = malloc(3 * sizeof(int));
    if (a == NULL) return NULL;
    a[0]=1; a[1]=2; a[2]=3;
    return a;
}
/* 修法 2：由呼叫端提供緩衝區，函式只填值 */
void fill_array(int *out) { out[0]=1; out[1]=2; out[2]=3; }
```

檢誤

- 只說「會出錯」沒講出「懸空指標 / 函式返回後 stack 回收」。
- 修法 1 忘了在呼叫端 `free`（變成洩漏）。

Q5 解答（手寫函式｜20 分）

解答

```c
int *make_squares(int n) {
    int *a = malloc(n * sizeof(int));
    if (a == NULL) return NULL;       /* 配置失敗處理 */
    for (int i = 0; i < n; i++)
        a[i] = i * i;
    return a;
}
```

呼叫端用完這塊記憶體後，必須 `free` 它（並把指標設 `NULL`），否則洩漏。

檢誤

- `malloc(n)` 漏乘 `sizeof(int)`。
- 沒檢查 `NULL`。
- 沒交代呼叫端要 `free`。

Q6 解答（抓 bug｜15 分）

解答

- **(1) use-after-free**：`free(p)` 後又 `*p = 5`。修：把寫入移到 `free` 之前；或先用完再 `free`。
- **(2) double-free**：同一塊 `free` 兩次。修：移除第二個 `free`；並在 `free` 後 `q = NULL`（`free(NULL)` 安全）。
- **(3) 記憶體洩漏**：第一塊 `malloc` 的指標被第二次 `malloc` 覆蓋，再也 `free` 不到。修：覆蓋前先 `free(r)`，或改用 `realloc(r, …)`。

檢誤

- (1)(2) 名稱互換（use-after-free vs double-free）。
- (3) 只說「沒 free」沒指出是「指標被覆蓋導致無法 free」。