

# 複習測驗卷 (答案卷)

時限 55 分鐘 · 總分 100

本卷為答案與評分要點。

## § 1 解答

### Q1 解答 (預測輸出 | 15 分)

解答

```
2
B
9
```

`67-65=2` ; `'A'+1 = 66 = 'B'` (`%c` 印字元) ; `'9'-'0' = 57-48 = 9`。

檢誤

· 第二行答 66 (用 `%c` 要印字元 B，不是數字)。

### Q2 解答 (預測輸出 | 15 分)

解答

印 `-2147483648`。 `int` 到最大值 2147483647 再 +1 發生**整數溢位**，繞回最小負數 (不會 crash)。需要更大的數用 `long long`。

### Q3 解答 (概念 | 15 分)

#### 解答

浮點數用二進位近似儲存，`0.1`、`0.2` 都無法精確表示，`0.1 + 0.2` 實際是 `0.30000000000000004`，與 `0.3` 的近似值不同，所以 `==` 為 `false`。正確做法：比較差值是否小於容差，`fabs(a - b) < 1e-9`。

### Q4 解答 (抓 bug | 20 分)

#### 解答

`sum` 沒初始化，初值是記憶體殘留的**垃圾值**（未定義行為），`sum += i` 從垃圾值開始累加，結果不可預測、每次可能不同。修正：`int sum = 0;`，修好後輸出 `15`。

#### 檢誤

- 以為區域變數會自動歸零（不會；只有全域/靜態變數才會）。

### Q5 解答 (手寫 | 20 分)

#### 解答

```
#include <stdio.h>
int main(void) {
    double c;
    scanf("%lf", &c);
    printf("%.1f\n", c * 1.8 + 32);
    return 0;
}
```

`double` 用 `%lf` 讀、`%.1f` 印。用 `int` 會掉小數。

#### 檢誤

- `scanf` 讀 `double` 用了 `%f`（要 `%lf`）；用 `int` 存溫度。

## Q6 解答 (概念 | 15 分)

---

### 解答

(a) `const` 是有型態的變數、有型態檢查、改它會編譯錯，**推薦**；`#define` 是前處理器的純文字替換、無型態。(b) 未初始化的變數是記憶體殘留的垃圾值，使用它是未定義行為（結果不可預測、難 debug），所以宣告時就要給初值。