

Solution of Quiz 01

1. 關於寫程式的步驟下列順序何者正確？

①執行程式 ②對程式碼進行編譯 ③儲存程式碼 ④用純文字編輯器撰寫程式碼 ⑤產生執行檔

- a) ①→②→③→④→⑤
- b) ④→②→③→⑤→①
- c) ④→③→②→⑤→①
- d) ②→⑤→①→③→④

2. 下列敘述何者錯誤？

- a) 編譯器(compiler)就像翻譯人員，將你的程式碼翻譯成電腦看得懂得機器碼。
- b) 使用 MS Word 編寫程式碼，儲存後將副檔名改成 .cpp，也可以對其進行編譯。
- c) 有些語言的程式碼不必經過編譯(compile)也可以執行，但 C++ 語言的程式碼須經過編譯才可以執行。
- d) 機器碼只由 0 跟 1 所構成(2 進位)。
- e) 理論上，若我們讀得懂機器碼，也可以直接編寫機器碼給電腦執行。

Ans: 因為 MS Word 非純文字編輯器。

3. 下列敘述何者錯誤？

- a) 編譯器是一個可執行檔，以你寫的程式碼當輸入，輸出為機器碼所構成的可執行檔。
- b) g++ 編譯器也可以將 C 語言編譯為可執行檔。
- c) g++ 編譯器也可以將 JAVA 語言編譯為可執行檔。
- d) 我們也可以撰寫 C++ 程式碼，編譯產生出一個 C++ 編譯器。

4. 下列敘述何者錯誤？

- a) 若你的程式碼無法被編譯(有錯誤訊息)，代表你的程式碼有 bug。
- b) 若你的程式碼可以被編譯，並已產生出可執行檔，代表你的程式碼一定沒有 bug。
- c) 如果程式碼執行的結果非你所預期的結果，代表程式碼可能有 bug。
- d) 若沒有良好的編程風格(coding style)，會增加 debug 的難度。

Ans: 可能有語意錯誤，程式碼可編譯，但執行後產生錯誤。

5. 下列敘述何者錯誤？

- a) 相同語言不同的程式碼，可能會有相同的功能(相同輸入，產生相同結果)。
- b) 不同語言不同的程式碼，可能會有相同的功能(相同輸入，產生相同結果)。
- c) 寫程式至少要有「純文字編輯器」與「編譯器」。
- d) 像電腦裡的記事本、MS Power Point，都是純文字編輯器

Ans: MS Power Point 非純文字編輯器。

6. 下列哪一程式碼可以印出一個雙引號(")?

- a) `cout << "";`
- b) `cout << '";`
- c) `cout << "";`
- d) `cout >> "";`

Ans: 利用跳脫字元 '\',

`cout << "\";`

或

`cout << "\";`

也都可以，但 `cout << "";` 不可，因為它會當成前 2 個"中夾一個字串(空字串)，但第 3 個"多出來不知道要幹麻。

7. 請問右方程式碼會在螢幕上印出什麼?

```
cout << "this /*comment*/ is fun\n";
```

- a) `this /*comment*/ is fun\n`
- b) `this /*comment*/ is fun`
- c) `this is fun`
- d) `this comment is fun`
- e) `this /comment/ is fun`

Ans: 字串(兩個"內夾的東西)裡的文字(字元)是一般文字，不會有註解功能(字串裡每個字就是字，除了逃脫字元 '\', 會讓它後一個字有特殊功能變沒特殊功能，或沒特殊功能變有特殊功能。

8. 請問右方程式碼編譯並執行後會在螢幕上印出什麼?

a)

```
1
11
121
1331
```

b)

```
1
11
121
1331
```

c)

```
1
11
121
1331
```

d)

```
1\n
11\n
121\n
1331\n
```

```
#include <iostream>
using namespace std;

int main(){
    cout << "1\n";
    cout << "11\n";
    cout << "121\n";
    cout << "1331\n";
}
```

Ans: 程式碼內部的空白、換行、tab(縮排)，是給人看的，對電腦無影響。  
要印出 b) 可以使用：

```
#include <iostream>
using namespace std;

int main(){
    cout << "  1\n";
    cout << " 11\n";
    cout << " 121\n";
    cout << "1331\n";
}
```

c) 請仿照 b) 一樣自己算空白數寫在兩個雙引號內

```
#include <iostream>
using namespace std;

int main(){
    cout << "   1  \n";
    cout << "  1 1  \n";
    cout << " 1 2 1 \n";
    cout << "1 3 3 1\n";
}
```

d) 可參照以下，利用逃脫字元\使它後面接的\失去特殊功能變成一般的\，連帶的也使 n 無特殊功能。

```
#include <iostream>
using namespace std;

int main(){
    cout << "  1\\n";
    cout << " 11\\n";
    cout << " 121\\n";
    cout << "1331\\n";
}
```

9. 如何在螢幕上印出第 7 題右方的程式碼？填寫後用電腦執行看看，並驗證答案是否正確。

```
cout << "cout << \"this /*comment*/ is fun\\n\\n\";\\n\";
```

Ans: 通常 cout 後面要加一個換行字元('\n')，未加換行字元的 cout 通常是 prompt (提示使用者要打字的消息，參照第一份投影片「導論」第 42 頁的程式碼：

```
cout << "Please input a number: "; )。
```

10. 請問右方程式碼編譯並執行後會在螢幕上印出什麼？

- a) Question 10
- b) 什麼都沒有
- c) 10
- d) 編譯錯誤

```
#include <iostream>
using namespace std;
int main(){
    cout << /*"Question"*/ 10;
}
```

Ans: /\*"Question"\*/ 是註解，所以

```
cout << /*"Question"*/ 10; 等同 cout << 10;
```

(p.s.) main function 理論上最後要 return 0;，但因為太多人容易忘記了，所以若沒寫編譯器會自動視同 return 0，所以不會有編譯錯誤(頂多警告(warning))。

11. 在 C++ 程式碼裡，像 `#include <iostream>` 的一行被稱作？

- a) Preprocessor directives
- b) Shell commands
- c) Statement blocks
- d) Machine instructions
- e) Program executables

12. 編譯式語言 (compiled languages) 與直譯式語言 (interpreted languages) 做比較，前者優勢在於？

- a) 編譯式語言比較 friendly 因為它可以一次編譯並執行一行。
- b) 編譯式語言有註解，直譯式語言則沒有。
- c) 編譯式語言對一個程式比較有 global view 且比較能優化程式。
- d) 編譯式語言可以解決的問題比直譯式語言多。
- e) 編譯式語言用 2 補數表示法 (2's complement) 表示整數，直譯式語言則是用 1 補數 (1's complement)。

Ans: a) 那是直譯式 b) 都有 d) 一樣多 e) 都是 2 補數

13. 一般編譯器 3 stages 是？

- a) cout, cin, and main
- b) ls, cp, and mv
- c) shell, Cygwin, and vi
- d) preprocessor, compiler, and linker
- e) directives, expressions, and statements

14. 考慮最左邊的 bit 是 MSB (most significant bit),  $\boxed{11111111_2}$  在下列表示法各代表哪一個十進位 (decimal) 整數?

a) 正整數 (Unsigned integer): 255

$$2^0 + 2^1 + 2^2 + \dots + 2^7 = 2^8 - 1 = 255$$

b) 符號帶大小 (Sign-magnitude): -127

$$\text{最左邊的 } 1 \text{ 是代表負數, 後面 } 1111111_2 \text{ 值} = 2^0 + 2^1 + 2^2 + \dots + 2^6 = 2^7 - 1 = 127$$

c) 2 補數 (2's complement): -1

$$-11111111_2 = \sim 11111111_2 + 1 = 00000000_2 + 1 = 1$$

d) 1 補數 (1's complement): -0

$$-11111111_2 = \sim 11111111_2 = 00000000_2 = 0$$

15. 以下 2 進位運算後的答案是 (以二進位表示)?

a)  $10110110_2 \& 11111111_2 = 10110110_2$

$$0 \& 1 = 0 \text{ and } 1 \& 1 = 1, \text{ so } x \& 1 = x$$

b)  $10110110_2 \mid 11111111_2 = 11111111_2$

$$0 \mid 1 = 1 \text{ and } 1 \mid 1 = 1, \text{ so } x \mid 1 = 1$$

c)  $10110110_2 \& 0 = 0$

$$0 \& 0 = 0 \text{ and } 1 \& 0 = 0, \text{ so } x \& 0 = 0$$

d)  $10110110_2 \mid 0 = 10110110_2$

$$0 \mid 0 = 0 \text{ and } 1 \mid 0 = 1, \text{ so } x \mid 0 = x$$

e)  $10110110_2 \& 1111_2 = 00000110_2$

$$1111_2 = 00001111_2$$

f)  $10110110_2 \mid 1111_2 = 10111111_2$

$$1111_2 = 00001111_2$$

16. 將 16-bit 的 -8 做 sign extension 至 32-bit 則值變成: -8

挑戰題:

1.  $0x3C5A6B9D \& 0x7F = 0x1D$

$$0x3C5A6B9D \& 0x0000007F = 0x3C5A6B(1001 \mid 1101) \& 0x000000(0111 \mid 1111) = 0x000000(0001 \mid 1101) = 0x0000001D$$

2.  $0x3C5A6B9D \& 0xFFFFFC00 = 0x3C5A6800$

$$0x3C5A6B9D \& 0xFFFFFC00 = 0x3C5A6(1011)9D \& 0xFFFF(1100)00 = 0x3C5A6(1000)00 = 0x3C5A6800$$

3.  $\sim 0 = -1$

$$\sim 0 = \sim 000\dots 0 \text{ (32 個 } 0) = 111\dots 1 \text{ (32 個 } 1) = -1 \text{ (看第 14 題的 c)}$$

4.  $\sim -1 = 0$

$$\sim -1 = \sim 111\dots 1 \text{ (32 個 } 1) = 000\dots 0 \text{ (32 個 } 0) = 0$$

5.  $\sim 3 = -4$

$\sim 3 = \sim 000\dots 011$  (30 個 0) =  $111\dots 100$  (30 個 1)

let  $x = 111\dots 100$  (30 個 1)

$-x = \sim x + 1 = 000\dots 011$  (30 個 0) + 1 =  $000\dots 0100 = 4$

so  $x = -4$

