

臺北市立建國高級中學 105 學年度資訊科培訓  
第二次模擬賽（喵難度）筆試試卷

請作答在答案卷上。

甲、選擇題（每題 3 分，共 30 分）

說明：第 1 題至第 10 題，每題選出一個最適當的選項，答錯不倒扣。

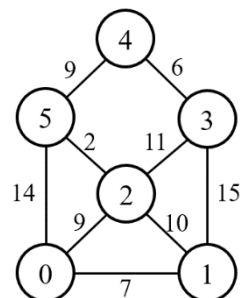
- 下列何者壓縮格式可以屬於無失真壓縮？  
(A) PNG                      (B) RAR                      (C) TIF                      (D) 以上皆是
- 請問下列哪種排序演算法在輸入資料很少時有非常高的效率？  
(A) 插入排序 (insertion sort)                      (B) 快速排序 (quicksort)  
(C) 泡沫排序 (bubble sort)                      (D) 堆排序 (heapsort)
- 已知一字串  $A = \text{"abcabdabcabc"}$ ，請問其每一個位置的 Z 函數值加起來是多少？（此處 Z 函數指的是 Gusfield's Algorithm 中的 Z 函數。）  
(A) 8                      (B) 9                      (C) 10                      (D) 12
- 假如一個演算法的複雜度可以表示成  $T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^3 \log^2 n)$ ，問這個演算法的複雜度是？  
(A)  $T(n) = \Theta(n^3 \log^2 n)$                       (B)  $T(n) = \Theta(n^3 \log^3 n)$   
(C)  $T(n) = \Theta(n^{\log_2 7})$                       (D)  $T(n) = \Theta(n^{\log_2 7} \log^2 n)$
- 下列哪個數轉換成二進制時會出現非零的循環小數？  
(A) 13.125                      (B) 84.25                      (C) 712.45                      (D) 16.75
- 已知 IEEE 754 規範的 128 位元浮點數當中，由高位到低位依序為「符號位」1 位元、「指數域」15 位元、「尾數域」112 位元。請問下列關於這種浮點數的敘述，何者錯誤？  
(A) 可以儲存的最大數（除了 inf 以外）大約是  $2^{2^{14}}$  左右  
(B) 在二進位之下，其有效位數共有 112 位  
(C) 在有些電腦上，使用 long double 會是這種型別  
(D) 如果最高的 16 個位元是 1，其他都是 0，那代表的數是 -inf
- 下列哪一對硬體的接頭無法相容？  
(A) USB 3.0 Type A / USB 2.0 Type A                      (B) VGA 端子 / DE-15  
(C) HDMI 1.4 Type A / HDMI 1.3 Type A                      (D) DDR3 SDRAM / DDR2 SDRAM

8. 2008 年發表的一種排序演算法「塊排序」(block sort)，它使用平方分割的概念改良  $\square\square\square\square$ ，是一個混合 (hybrid)、穩定 (stable)、原地 (in-place)、有適應性 (adaptive) 且最差複雜度為  $O(n \log n)$  的演算法。根據以上敘述，下列何者正確？
- (A) `<algorithm>` 中 `std::sort` 函數符合混合、原地及最差複雜度  $O(n \log n)$  三個條件
- (B) 此演算法的最佳情況複雜度是  $O(n)$ ，發生在輸入已經排序好的時候
- (C)  $\square\square\square\square$  有可能填入「快速排序」
- (D) 合併排序 (merge sort) 符合穩定、適應性兩個條件
9. 近來「固態硬碟」在市面上愈來愈普及。關於固態硬碟的敘述，何者錯誤？
- (A) 標示為 128GB 容量的固態硬碟，實際有 128GiB 的容量，但多餘的這些容量用作預留空間以作為垃圾收回或資料保護之用，無法實際使用
- (B) 相對於傳統硬碟，固態硬碟的缺點是極難救回已損毀的資料
- (C) 相對於傳統硬碟，固態硬碟的優點是寫入更快且可寫入的次數更多
- (D) 固態硬碟多數使用 NAND Flash 儲存資料
10. C++ 語言由 C 語言衍生而來，因此 C 和 C++ 程式碼有一定的相似程度，但是也有不相容的情況。下列何者程式碼在 C90 和 C++11 都能正常編譯？(假設所需要的標頭檔皆已被引入)
- (A) `int* a = malloc(sizeof(int) * 5);`
- (B) `int a, b, c, d, e; e = a ? b : c = d;`
- (C) `typedef struct { int b; } Type; Type a; //custom struct`
- (D) `int a = 4; a xor_eq 12;`

乙、填充題 (每題 4 分，共 24 分)

說明：第 11 題至第 16 題，每題選出一個最適當的選項，答錯不倒扣。

11. 使用 Dijkstra's Algorithm 求右圖編號為 0 的點到所有點的最短路徑，則編號為\_\_\_\_\_點的最短路徑恰好被成功更新 (鬆弛) 兩次。(可能不只一個點)



12. 一個 `int` 變數的值原本是 10，變數名稱為 A，在執行下列程式碼後，其值變為\_\_\_\_\_。
- ```
A = 6 + A >> 1 + 1 | 3 ^ A <= 6 == 4 < 2;
```

13. 中國在 2016 年 6 月測試了花費 18 億人民幣建造的超級電腦「神威·太湖之光」(Sunway TaihuLight)，其運算效能達到了 93 PFLOPS，是當前最快的電腦。假設這臺電腦的 CPU 時脈為 1.5GHz，總共有 1000 萬個核心，每一個週期可以進行 8 次浮點數運算，則其效率 (實際效率 / 理論效率 \* 100%) 為\_\_\_\_\_%。

14. 紅黑樹是一種應用廣泛的二元搜尋樹，如 C++ 中 `set`、`map` 便是實作這種資料結構。它除了一般的二元搜尋樹性質之外，有以下三個要求：
- 一、所有節點不是紅色就是黑色。
  - 二、根和葉節點（空指針，不儲存任何資料）都是黑色。
  - 三、每個從根到葉節點的簡單路徑都必須有相同數目的黑色節點，且不能出現連續兩個紅色節點。
- 請問一個高度（從根到葉經過邊數的最大值）為 6 的紅黑樹，最少有 \_\_\_\_\_ 個內部節點（不是葉子的節點，含根）。
15. 在 OSI 七層模型（開放式系統互聯通訊參考模型）當中，表現層（`presentation layer`）和會議層（`session layer`）較少被使用，原因是某些 \_\_\_\_\_ 層的協定具有類似的功能。（填中文名稱。）
16. 在 C++11 裡面寫 `std::unordered_set<std::pair<int, int>> a;` 會導致編譯錯誤，原因是因為在 `<functional>` 標頭檔裡沒有 \_\_\_\_\_ 的定義。

丙、程式模擬題（每題 5 分，共 45 分）

說明：第 17 到 25 題，每一題會有一段 C++11 程式碼，請在下方的空格寫出程式執行後輸出到標準輸出（standard output）的結果，並在右方空格寫出額外訊息，如下所列：

1. 如果無法通過編譯，請寫「CE」（編譯失敗），後面接著寫第一個編譯錯誤的行數。保證不會有拼字錯誤或未引入標頭檔而導致的編譯失敗。
2. 如果在編譯器不優化的情形之下，在不同電腦、時間、編譯器上執行的結果仍可能不同，請寫「UB」（未定義行為），並在下方寫出一種可能發生的輸出結果。已知 `char`, `short`, `int`, `long`, `size_t` 分別是 1, 2, 4, 8, 8 位元組的整數型別。
3. 如果程式並不會輸出任何東西，請寫「空白」。

題目可能有額外要求或說明，詳見題目。全對才給分，答錯不倒扣。

範例：

S1.

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     cout << "\ " << endl;
5 }
```

作答範例：

| S1. | CE 4 |
|-----|------|
| 1   |      |
| 2   |      |
| 3   |      |

（因為在此例中雙引號不左右對稱，故編譯錯誤。）

提示：小心不要被惡意的排版騙了！

17.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << ((4, 2) + (4, 5)) << endl;
6 }
```

18. 保證本題可以順利編譯、執行，並且不會引發未定義行為。

註：pair<iter, iter> std::minmax\_element(iter first, iter last)位於<algorithm>標頭檔內，回傳的是一個 pair，first 和 second 分別是指向[first, last)中最小元素和最大元素的迭代器。預設使用<（小於）運算子。

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 int cnt = 0;
6
7 struct Type {
8     int a;
9     bool operator<(const Type& rhs) const {
10         cnt++;
11         return (a < rhs.a);
12     }
13 } Data[100];
14
15 int main() {
16     for (int i = 0; i < 100; i++) Data[i].a = i;
17     minmax_element(Data, Data + 100);
18     cout << cnt << endl;
19 }
```

19.

```
1 #include <iostream>
2 #include <vector>
3 #include <bitset>
4 using namespace std;
5
6 int main() {
7     vector<int> A(10);
8     bitset<10> B(10);
9     try {
10         cout << A[10] << endl;
11         cout << B.set(10) << endl;
12     }
13     catch (out_of_range a) {
14         cout << A[0] << B << endl;
15     }
16 }
```

20.

```
1 #include <iostream>
2 using namespace std;
3
4 struct Type {
5     int a;
6     Type& operator+=(Type&& b) {
7         a += b.a; cout << a << endl;
8         return (*this);
9     }
10    Type operator+(const Type& b) {
11        return (Type){a + b.a};
12    }
13 };
14
15 int main() {
16     Type A = (Type){2};
17     (A += (Type){4} + A) += A + A;
18     A += A;
19 }
```

21. 已知此程式必定在 Little-Endian 的環境執行。

```
1 #include <iostream>
2 using namespace std;
3
4 unsigned int d[4] = {0xd5acf2ee, 0x13120100, 0xa1b2c3d4, 0};
5
6 int main() {
7     cout << *(short*)((char*)&d + 3) << endl;
8 }
```

22. 保證本題可以順利編譯。

```
1 #include <iostream>
2 using namespace std;
3
4 int f(int x) { return x<<2 + 1; }
5 int g(int x) { return x ^ 8; }
6
7 int main() {
8     int k = 10;
9     cout << f(++k) + g(++k) << endl;
10 }
```

23.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int k = 10;
6     if (k > 15)
7         if (k > 5) cout << "A\n";
8         else cout << "B\n";
9 }
```

24. 保證本題可以順利編譯。

```
1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 int d[4] = {2, 3, 4, 5};
6
7 int main() {
8     sort(d, d + 4, [](int a, int b) {
9         return (a ^ b * 2 + 1) % 13 < 6;});
10    cout << d[0] << d[1] << d[2] << d[3] << endl;
11 }
```

25. (註：本題改編自 TOI 入營考考題。)

```
1 #include <iostream>
2 using namespace std;
3
4 int a;
5
6 int f(int x)
7 {
8     if (x == 0) a = 1;
9     else a *= f(x - 1);
10 }
11
12 int main() {
13     cout << f(5) << endl;
14 }
```

丁、往事題(?) (每題 1 分，共 1 分)

26. 上次模擬賽哪一題用到二分搜? \_\_\_\_\_  
(填第幾題、TIOJ 題號或題目標題)

臺北市立建國高級中學 105 學年度資訊科培訓  
第二次模擬賽 (喵難度) 筆試答案卷

甲、選擇題 (每題 3 分, 共 30 分)

|    |    |    |    |     |
|----|----|----|----|-----|
| 1. | 2. | 3. | 4. | 5.  |
| 6. | 7. | 8. | 9. | 10. |

乙、填充題 (每題 4 分, 共 24 分)

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 11. | 12. | 13. | 14. | 15. |
| 16. |     |     |     |     |

丙、程式模擬題 (每題 5 分, 共 45 分)

|     |  |     |  |     |  |
|-----|--|-----|--|-----|--|
| 17. |  | 18. |  | 19. |  |
| 1   |  | 1   |  | 1   |  |
| 2   |  | 2   |  | 2   |  |
| 3   |  | 3   |  | 3   |  |
| 20. |  | 21. |  | 22. |  |
| 1   |  | 1   |  | 1   |  |
| 2   |  | 2   |  | 2   |  |
| 3   |  | 3   |  | 3   |  |
| 23. |  | 24. |  | 25. |  |
| 1   |  | 1   |  | 1   |  |
| 2   |  | 2   |  | 2   |  |
| 3   |  | 3   |  | 3   |  |

丁、往事題(?) (每題 1 分, 共 1 分)

|     |
|-----|
| 26. |
|-----|