

CH08

是非題

- (○) 1. 資料機密性通常透過資料加密來達成。
- (○) 2. 對稱式加密演算法的執行效率，一般而言，較非對稱式加密演算法來得好。
- (×) 3. 在密碼長度相同的情況下，非對稱式的加密演算法較對稱式加密演算法來得安全。
- (○) 4. 使用非對稱式金鑰演算法進行加密，只有擁有私密金鑰的使用者，可以順利地進行解密。
- (×) 5. 網站的網址只要是使用 https 的協定傳輸，就一定是安全無虞的。
- (×) 6. 手機所使用的 A5 演算法，是一種對稱式的區塊加密演算法。
- (○) 7. RSA 演算法是基於因數分解的困難度設計而成的。
- (×) 8. 只要不知道密碼，就永遠都無法得知被加密保護的密文。
- (○) 9. 憑證 (certificate) 的目的是用以證明公開金鑰的擁有者以及其背書者是
否正確。
- (×) 10. 雜湊函數的輸出與輸入內容的長度成比例。

選擇題

- (A) 1. 下列何種對稱式演算法是美國國家標準技術局自 2001 年起採用的加密標準?
- (A)AES (B)BES (C)CES (D)DES
- (C) 2. 若要在公開場合交換密碼,我們可使用下列何種演算法進行交換?
- (A)RSA (B)ElGamal (C)Diffie-Hellman (D)BlowFish
- (B) 3. 阻斷服務攻擊(DoS)的進階版 DDoS,其字首 D 的意義為何?
- (A)Daniel (B)Distributed (C)Denial (D)Distributor
- (D) 4. 下列何者不是常見的無線網路傳輸相關的加密機制?
- (A)WEP (B)TKIP (C)WPA (D)WAP
- (C) 5. 下列哪種應用和非對稱式加解密演算法無關?
- (A)RSA (B)DSA (C)DES (D)ElGamal
- (B) 6. RSA 演算法是基於計算何種問題的困難度設計而成?
- (A)離散對數 (B)因數分解 (C)二次剩餘 (D)橢圓曲線

填充題

1. 基於效率考量,數位簽章通常不直接對內容進行簽章,而是針對內容的 雜湊值 進行簽章。
2. 我們可以使用 雜湊函數 來檢查資料是否遭到修改。
3. 駭客自製介面精美的偽冒網站,以吸引使用者提供其帳號密碼等個人資訊。這種攻擊我們稱為 (網路)釣魚。
4. 表面上沒有惡意,卻暗地裡在電腦主機上開啟後門的程式,我們常常稱其為 特洛伊木馬 (Trojan horse)。

5. 我們可以透過 加密 的方式，避免資料在網路傳輸時遭到監聽。
6. 目前常見的無線網路連線常用的安全加密機制為 WEP 以及 WPA。

簡答題

1. 請列舉 3 種常見的對稱式金鑰加密演算法。

【詳解】

DES、AES、IDEA、RC5 等。

2. 區塊加密的對稱式加密演算法常常需要配合操作模式如 CBC、CTR 等運作，其主要原因為何？

【詳解】

其主要原因是避免使用相同的密碼與相同的演算法，對相同的資料進行加密時，產生相同的密文。透過初始向量 (IV)、密文以及加密資料的 XOR 運算，可以提升加密資料安全性。

3. 請列舉 2 種常見的雜湊演算法。

【詳解】

MD5、SHA1、SHA256 等。

4. 我們在建置高可用性的系統時，常常使用「心跳」(heartbeat) 機制。請簡述心跳機制的做法。

【詳解】

心跳機制讓二個系統之間，互相偵測另一個系統是否還是在正常運作中。簡單的說，心跳機制就是讓二台主機之間定期交換一個特定的探測訊息。如果其中一方發現另一個系統沒有回應時，就可以判斷系統異常而啟動備援，接手工作的機制。

CH09

是非題

1. C 語言寫出來的程式，比組合語言寫出來的程式，可攜性較低。

【解答】 ×

2. LISP 常被用來撰寫人工智慧的應用。

【解答】 ○

3. 在類別中，我們可以定義資料和行為。

【解答】 ○

4. 我們常利用指標，來表示不確定大小的資料。

【解答】 ○

5. 在 PASCAL 裡，我們利用 repeat 指令來表示執行固定次數的迴圈。

【解答】 ×

選擇題

1. C 語言是屬於哪一種語言：

(A)高階語言 (B)低階語言 (C)自然語言 (D)組合語言

【解答】 (A)

2. 下列何者為最早提出來的高階語言：

(A)JAVA (B)C (C)FORTRAN (D)BASIC

【解答】 (C)

3. 下列何者為物件導向程式語言：

(A)C++ (B)PROLOG (C)ADA (D)PASCAL

【解答】 (A)

4. 下面哪一項資料型態，是處理一序列具有相同型態的資料：

(A)字元 (B)陣列 (C)結構 (D)浮點數

【解答】 (B)

5. 在呼叫一個程序時，若是直接把真實參數的值，指定給正式參數，則這種方法我們稱作：

(A)以值傳遞 (B)以位址傳遞 (C)以名傳遞 (D)以上皆非

【解答】 (A)

填充題

1. 專為商業資料處理而開發設計出來的語言，為 _____ 程式語言。

【解答】 COBOL

2. 一般程式語言提供的數字型態，包含了 _____、_____、_____和 _____等。

【解答】 整數、長整數、浮點數、雙精準數

3. if 指令提供了邏輯判斷式的寫法，也就是，如果 _____指令後面接著的運算式被判斷為真，則程式會繼續執行 _____指令後面的運算式，否則執行 _____指令後面的運算式。

【解答】 IF、Then、Else

4. PASCAL 裡的"begin"指令和"end"指令，對應到 C 語言裡的 _____符號和 _____符號。

【解答】 { 符號和 }

5. 在流程圖裡，用以表示決策的運算式，是用 _____表示；用以表示計算的敘述式，是用 _____表示。

【解答】 菱形框、長方框

問答題

1. 宣告一個結構，來表示公司裡一個員工的相關資料。

【詳解】

以下宣告一個員工，具有姓名、地址、職稱、薪水等資料。

```
struct employee {  
    char (6) name;  
    char (20) address;  
    char (10) title;  
    int salary;  
};
```

2. 利用 C 語言裡的"while"指令，計算整數 1 到 100 的和。

【詳解】

```

i = 1; x = 0;
while ( i <= 100 )
{
    x = x + i;
    i = i + 1;
}

```

3. 列出定義一個程序時，所需要提供的四項資訊。

【詳解】

程序在定義時，必須提供下列資訊：

1. 程序名稱
2. 程序本體，含變數宣告和命令敘述
3. 正式參數（Formal parameter）宣告
4. 程序回傳的資料型態

4. 說明全域變數和局部變數的差別。

【詳解】

全域變數（Global variable）能被全部的程式碼使用到；而局部變數（Local variable）只能被一部分程式碼使用到，通常定義在程序中。

5. 請上網查詢有關 JAVA 程式語言的特性。

【詳解】

完整的資料可以由美國 Sun 公司所提供的網站 <http://java.sun.com> 所取得。

6. 列舉物件導向程式語言的特性。

【詳解】

具有封裝特性的物件，為程式的核心。

7. 撰寫一個程序"sum"，其中定義一個整數參數"n"，然後該程序會回傳"1"加到整數"n"的和。

【詳解】

```

int sum(int n)
{
    int i = 1, x = 0;
    while ( i <= n )
    {
        x = x + i;
        i = i + 1;
    }
    return(x);
}

```

8. 撰寫一個程序"sum"，其中定義一個整數參數"n"，然後該程序會回傳"1"加到整數"n"的和。但是必須判斷參數"n"是否為正整數，若小於 0 的話則程序直接回傳 0。

【詳解】

```
int sum(int n)
{
    int i = 1, x = 0;
    if (n < 0)
        return(0);
    while ( i <= n )
    {
        x = x + i;
        i = i + 1;
    }
    return(x);
}
```

9. 討論在什麼情況下，程序需要用到「以位址傳遞」的方式。

【詳解】

希望改變原本真實參數的值。

10. 上網查詢目前 COBOL 程式語言發展的近況。

【詳解】

相關的資料可由 <http://www.cobolportal.com> 查詢到。

CH10

一、是非題

1. 陣列裡元素的資料型態可以不同。

【解答】 ✕

2. 在程式執行時，陣列裡註標比較小的元素，會比註標大的元素更快拿到。

【解答】 ✕

3. 環狀佇列是採用「先進先出」的順序。

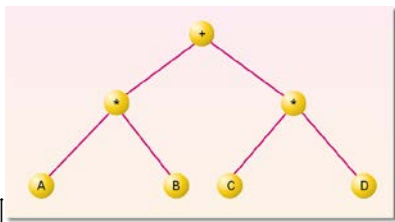
【解答】 ○

4. 環狀佇列裡宣告的每一個空間，都可以填入資料。

【解答】 ✕

5. 樹只有一個根節點。

【解答】 ○



6. 在圖中的二元樹，其樹高為 3。

【解答】 (○)

二、選擇題

1. 以下何者代表 C 語言裡的空指標：

(A)null (B)nil (C)empty (D)not

【解答】 (A)

2. 以下何者的邏輯順序和實體順序不一定相同：

(A)鏈結串列 (B)一維陣列 (C)二維陣列 (D)以上皆是

【解答】 (A)

3. 以下哪種資料結構是採用「後進先出」的順序：

(A)陣列 (B)佇列 (C)堆疊 (D)環狀佇列

【解答】 (C)

4. 從根節點到樹中所有葉節點的最長可能路徑，稱作樹的

(A)高度 (B)階層 (C)根節點 (D)葉節點

【解答】 (A)

5. 在二元樹的探訪順序中，先探訪父節點、再探訪左子節點、最後探訪右子節點，稱作

(A)前序法 (B)中序法 (C)後序法 (D)循序法

【解答】 (A)

三、填充題

1. 假設系統在記憶體裡記錄多維陣列的方法，是先從第一列開始，然後接著記錄第二列，這種方式稱作_____。

【解答】 以列為主

2. 根據 C 語言的語法，若在宣告一個變數時前面加上_____符號，則該變數就是指標變數。

【解答】 *

3. 所謂的二元樹，就是每一個節點最多只有_____個子節點。

【解答】 2

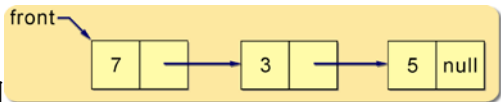
4. 將一個算數運算式以樹狀結構表示，此樹稱作_____。

【解答】 運算樹

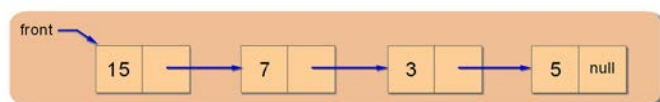
5. 在程序的本體中，又呼叫到自己本身，稱作_____程序。

【解答】 遞迴

四、問答題

1. 根據圖的鏈結串列，執行「insert(front, 15)」，畫出執行後的鏈結串列。

【詳解】



2. 利用第 7-3 節的堆疊宣告，改進程序"push"，要求在加入資料前，先判斷堆疊是否還有空位置。

【詳解】

```
void push (int data){
    if (top < 9)
    {
        top = top + 1;
        stack[top] = data;
    } else
    {
        printf("The stack is full.");
    }
}
```

3. 利用第 7-3 節的堆疊宣告，改進程序"pop"，要求在取出資料前，先判斷堆疊內是否有資料。

【詳解】

```
int pop( ){
    if (top >= 0)
    {
        top = top -1;
        return stack[top+1];
    }
    else
    {
        printf("The stack is empty.");
    }
}
```

4. 利用第 7-3 節的佇列宣告，改進程序"put"，要求在加入資料前，先判斷佇列是否還有空位置。

【詳解】

```
void put (int data){
    if (rear < 9)
    {
        rear = rear + 1;
        queue[rear] = data;
    } else
    {
        printf("The queue is full.");
    }
}
```

5. 利用第 7-3 節的佇列宣告，改進程序"get"，要求在取出資料前，先判斷佇列內是否有資料。

【詳解】

```
int get(){
    if (rear > front )
    {
        front = front +1;
        return queue[front];
    } else
    {
        printf("The queue is empty.");
    }
}
```

6. 討論何時使用陣列，何時使用鏈結串列。

【詳解】

如果資料不確定有多少，且時常動態增減，則較適宜使用鏈結串列。

7. 根據第 7-2 節 node 和 front 的定義，撰寫一個程序叫作 RemoveHead，把參數 front 指到的鏈結串列的第一個節點移除，然後回傳該節點所表示的資料（data）。

【詳解】

```
int RemoveHead(struct node *front)
{
    struct node *temp;
    temp = front;
    front = front->next;
    return(temp->data);
}
```

8. 討論何時使用堆疊，何時使用佇列。

【詳解】

如果我們希望先遇到的資料先處理，則使用佇列。反之，若希望先遇到的資料後處理，則使用堆疊。

9. 討論何時針對二元樹做後序法的探訪。

【詳解】

如果我們希望處理資料的順序，是先處理左子節點，接著是右節點，最後才處理父節點的話，則適用後序法。

CH11

填充題

1. 從 n 個數中找出最大數，最少要用 _____ 次比較。

【解答】 $n-1$

2. 給定 n 個數，請將它們由小排到大，稱為 _____ 問題。

【解答】 排序

3. _____ 排序法將數列切成兩部分：已排序數列及未排序數列，每次從未排序的數列中挑出最小的數，將它移到未排序數列的最前面。

【解答】 選擇

4. _____ 排序法將數列切成兩部分：已排序數列及未排序數列，每次將未排序數列中的第一個數，插入到已排序數列中，使得插入後的已排序數列仍然維持由小排到大的性質。

【解答】 插入

5. _____ 排序法將數列切成兩部分：已排序數列及未排序數列，每次從未排序數列中的最後一個數看起，如果它比前面的數小，則往前移，一直看到未排序數列的第一個數為止。

【解答】 泡沫

簡答題

1. 12 個金幣，有一個假的，只知和其他標準金幣重量不同，請用天平秤三次，就把假的金幣找出來，並確認它比較重或比較輕。（每次稱有三種可能性：大於、等於及小於，可用一個樹狀圖來描繪各種可能性）

【詳解】

有兩種方式，第一種可先四個和四個秤；第二種可先三個和三個秤。兩種方式展開的樹狀圖都可在秤三次情況下，找出假金幣。

2. 給定數列 23、12、58、85、72、98、13、37，請以課本介紹的兩個方法，找出其中的最大數及最小數，把你的作法記錄下來。

【詳解】

第一種方法逐一比較得最大數 98；第二種方法兩兩比較，98 會勝出。

3. 給定數列 23、12、58、85、72、98、13、37，請以課本介紹的方法，找出其中的最大數及第二大數，把你的作法記錄下來。

【詳解】

第一種方法逐一比較得最大數 98，再從剩下的 23、12、58、85、72、13、37 找出第二大數 85；
第二種方法兩兩比較得 98 最大，再從曾輸過 98 的 85、72、37 中找出第二大數 85。

4. 給定一個數列，請設計一個可找出前三大數的演算法。

【詳解】

兩兩比較找出最大數，再從曾輸過最大數的那些數中找出第二大數，再從曾輸過最大數和第二大數的數中找出第三大數。

5. 給定數列 23、12、58、85、72、98、13、37，請以「選擇排序法」將它由小排到大，記錄你的過程。

【詳解】

```
23  12  58  85  72  98  13  37
12 || 23  58  85  72  98  13  37
12  13 || 58  85  72  98  23  37
12  13  23 || 85  72  98  58  37
12  13  23  37 || 72  98  58  85
12  13  23  37  58 || 98  72  85
12  13  23  37  58  72 || 98  85
12  13  23  37  58  72  85  98
```

6. 給定數列 23、12、58、85、72、98、13、37，請以「插入排序法」將它由小排到大，記錄你的過程。

【詳解】

```
23  12  58  85  72  98  13  37
12  23 || 58  85  72  98  13  37
12  23  58 || 85  72  98  13  37
12  23  58  85 || 72  98  13  37
12  23  58  72  85 || 98  13  37
12  23  58  72  85  98 || 13  37
12  13  23  58  72  85  98 || 37
12  13  23  37  58  72  85  98
```

7. 給定數列 23、12、58、85、72、98、13、37，請以「泡沫排序法」將它由小排到大，記錄你的過程。

【詳解】

```
23 12 58 85 72 98 13 37
12 || 23 13 58 85 72 98 37
12 13 || 23 37 58 85 72 98
12 13 23 || 37 58 72 85 98
12 13 23 37 58 72 85 98
```

8. 給定數列 23、12、58、85、72、98、13、37，請以「快速排序法」將它由小排到大，記錄你的過程。

【詳解】

```
23 12 58 85 72 98 13 37
13 12 23 85 72 98 58 37
...
12 13 23 37 58 72 85 98
```

9. 給定數列 23、12、58、85、72、98、13、37，請以「合併排序法」(merge sort)將它由小排到大，記錄你的過程。(雖然本章沒介紹作法，但讀者可到圖書館找演算法相關書籍，以本章建立的基礎，應有辦法理解這個方法)

【詳解】

先以合併排序法排 23、12、58、85，得 12、23、58、85；再以合併排序法排 72、98、13、37 得 13、37、72、98。再將 12、23、58、85 及 13、37、72、98 依序合併得 12、13、23、37、58、72、85、98。

10. 給定數列 12、13、23、37、58、72、85、98，請以「二元搜尋法」找看看 85 在不在這數列中，也找找看 18 在不在這數列中，記錄你的過程。

【詳解】

找看看 85 在不在這數列中？先比較 85 和 37，因為 85 比較大，所以找後面部分；再比較 85 和 72，85 仍然比較大，再找後面部分；比較 85 和 85 時，回答 85 在這數列中。

找找看 18 在不在這數列中？先比較 18 和 37，因為 18 比較小，所以找前面部分；比較 18 和 13，18 比較大，所以比較後面的部分；此時只剩 23 和 18 比，並不相等，所以回答 18 不在這數列中。

11. 請解釋動態規劃技巧的解法三步驟。

【詳解】

動態規劃技巧有三個主要部分：遞迴關係 (recurrence relation) 用來定義最佳答案、列表式運算 (tabular computation) 用來找最佳答案的值及路徑迴溯 (traceback)，將最佳答案的組合列出。

12. 以 LCS 的方法，找 PROFESSOR 和 CONFESSION 這兩個序列的最長共同子序列。

【詳解】

OFESSO

13. 「旅行推銷員問題」和「小偷背包問題」這兩個問題，你有沒有想到好解法呢？

【詳解】

自由發揮。

14. 請計算 $1000n$ 、 $100n \log_2 n$ 、 $10n^2$ 、 n^3 及 2^n ，在 $n = 1$ 、 100 、 10000 及 1000000 時的值各為多少，把它們的大小關係列出來。

【詳解】

$1000n$ 剛開始會輸，但隨著 n 的增長，很快就會勝出； 2^n 很快就暴增了。

15. 已知 128 個金幣中有一假金幣（假的較輕），請問用天平最少秤幾次可以得知那一個是假金幣？

【詳解】

如果每次都盡可能平分成三堆，一定至少有兩堆金幣個數相同，把那相同個數的兩堆拿來秤，如果有一堆比較輕，那一堆一定包含那個假金幣，否則金幣就在沒秤的那一堆，再把包含假金幣的那堆依同樣作法盡可能平分成三堆做下去，…。128 個金幣平均分成三堆，三堆個數分別為 43、43、42，把那 43 個的兩堆拿來秤，如果一樣重，則假金幣在 42 個的那堆，否則比較輕的就包含假金幣，此時我們的問題大小已從 128 降到 42 或 43，比剛剛分兩堆的策略只降到 64 有效多了，所以這樣總共要秤幾次呢？最糟情況是：128、43、15、5、2，共 5 次，也就是

$\log_3^{128} = 5$ 次。