

DSE 資訊及通訊科技科

建議答案 (中文版本)

卷一 (甲部)

題號	答案	題號	答案
1.	B	21.	B
2.	D	22.	A
3.	A	23.	B
4.	D	24.	B
5.	A	25.	C
6.	C	26.	A
7.	A	27.	A
8.	A	28.	A
9.	C	29.	B
10.	D	30.	D
11.	B	31.	B
12.	D	32.	C
13.	D	33.	D
14.	A	34.	A
15.	C	35.	D
16.	B	36.	B
17.	A	37.	B
18.	B	38.	A
19.	D	39.	D
20.	C	40.	D

本文件內列有以下符號：

- | | |
|---|--------------------------------|
| × | 錯誤或不為接納的答案 |
| ■ | 答案缺少網底內的文字、數字或意念，但屬正確，仍可得分 |
| / | 顯示某個答案內可接受的不同版本或寫法 |
| + | 需要兩項資料，第一部分答案正確，才會考慮給予第二部分答案分數 |

卷一（乙部）

		分數	
1.	(a)	優點：聊天機器人能生成搜尋結果的摘要／能整合來自不同來源的資訊並生成總結，使結果更易理解／聊天機器人能提供更個人化的內容	1
		缺點：聊天機器人可能會出現「幻覺」問題，即錯誤地轉述資訊或生成不準確的內容	1
	(b)	聊天機器人的訓練數據是否來自權威／可靠的來源 聊天機器人的訓練數據是否為最新資訊，而非過時數據 訓練數據的數量 訓練數據的多樣性	1×2
2.	(a)	(i) 管理電腦與傳輸媒介（無線頻率）之間的數據傳輸／管理電腦與網絡之間的數據傳輸	1
		(ii) 將無線設備連接到有線網絡／提供無線連接以組成無線 LAN	1
		(iii) Wi-Fi 可能未經加密，導致被黑客竊聽的可能性較高／容易受到中間人攻擊 咖啡店可能會監控並記錄網絡流量，威脅數據私隱 較易連接到黑客設置的虛假 Wi-Fi 熱點	1×2
3.	(a)	浮點數	1
	(b)	Crop 23.33 Tree 25	1, 1
		① GROUP BY（2 行，正確的 Type） ① 全部正確	
	(c)	使用 SQL 提取數據時，數據庫比試算表更有效率（特別是處理大量數據時）／數據庫可以儲存和處理更大量的數據／可以實施更佳的數據驗證，例如：通過在數據庫中設置 ID 為主關鍵碼來確保 ID 的唯一性	1
4.	(a)	設上載頻寬為 x Mbps $(1.5 \times 1024 \times 1024 \times 1024 \times 8) \times 20 / x \times 1000 \times 1000 = 2 \times 60$ $x = 2147,484$ 最小上載頻寬 = 2147,484 Mbps	2
		① 正確公式 ① 正確答案（包括正確的單位）	
		✗ 沒有單位不會獲得答案分	
4	(b)	(i) <u>學生</u> 使用 <u>學生的私匙</u> 加密 影片 學校 / <u>老師</u> 使用 <u>學生的公匙</u> 解密 影片	1,1
		✗ 答案沒有提及使用是誰的匙 ✗ 答案沒有使用「加密」和「解密」等字眼	
		(ii) 黑客更難透過暴力破解攻擊破解加密系統	1

分數

✗ 需要更長時間解密

5. (a) 電腦 A

1 + 1

GPU：獨立的 GPU／顯示卡提升了並行處理性能，減少訓練模型的時間

CPU：時鐘頻率較高／核心的數量較多，提升處理數據或訓練指令的執行效率

RAM：較大的 RAM 容量／儲存空間，讓用戶能儲存更多機器學習所需的臨時數據

① 正確的電腦

① 正確的配置

✗ 只提及 CPU 較佳，而未說明具體規格（例如：時鐘頻率／核心數量）

✗ 電腦 B（頻寬較高／儲存空間較大）

(b) 電腦 B

1 + 1

連接埠：USB 3.0 的數據傳輸率高於 USB 2.0，提升了從相機傳輸視頻到電腦的速度

網絡(有線)：有線連接（2 Gbps LAN）的頻寬較高，這提升上傳視頻到互聯網的數據傳輸速率

網絡(無線)：採用更新的 Wi-Fi 標準 (802.11ax) 無線連接頻寬通常較高，這提升上傳視頻的數據傳輸速率

① 正確的電腦

① 正確的配置

分數

6. 5 6 9 6 1 4
i = 0

5 6 9 6 1 4
i = 1

5 6 9 6 1 4
i = 2

每個錯誤 -1 分
7. (a) =SUMIF(Sales!\$D\$2:\$D\$7, "<="&Total!A2, Sales!\$D\$2:\$D\$7) 3
① ① ① ①
- 符合任何以下兩項：
① 正確的函數名稱 (SUMIF)
① 正確的儲存格範圍 (D\$2:D\$7)
① 正確的跨表參照 (Sales!)
① "<="&
- 第三分：
① 全部正確
- (b) 列：TYPE 1
值： 1
TYPE / ArtworkID / SalesPrice 的 COUNT
TYPE / ArtworkID / SalesPrice 的計數
TYPE / ArtworkID / SalesPrice 的項目計數
(其他合理的答案)
- (c) (i) 桌上型電腦與流動電話的屏幕尺寸不同／不同的長寬比／不同的輸入方式／不同的互聯網連接速度 1
(ii) 圖像的檔案格式／圖像的解像度／圖像的長寬比 1
- (d) 提供用戶介面／ 1×2
資源管理 (記憶體管理／網絡管理／設備管理)
- (e) 優點：節省儲存空間／減少傳輸檔案時的傳輸時間 2
× 數據傳輸速率較高

缺點：她的同事在打開檔案前需要先解壓縮，這需要更多時間來開啟試算表檔案／她的同事可能需要安裝額外的軟件來解壓縮檔案

× 與有損壓縮相關的答案 (例如：質素下降)

		分數
8. (a)	<p>可以提供更多資訊（例如：不同語言的食物標籤／營養標籤） 可以提供多媒體／視頻（例如：關於其他飲品的廣告視頻） 資訊較易作出更新</p> <p>× 成本較低（沒有指明具體成本類型，例如「印刷成本」） × 更環保</p>	1×2
(b)	<p>整數。銷售飲品的數量必須是離散值，不能是小數。</p> <p>× 答案正確但沒有解釋 × 浮點數 × 數值（數值可以是整數或浮點數） × 陣列（陣列是數據結構，而非數據類型）</p>	1
(c) (i)	<p>它判斷陣列 sales 是否以升序排列／它判斷每日的銷售額是否低於前一天 它判斷陣列 sales 是否並非以升序排列／它判斷每日銷售額和前一日相比是否並非遞減</p> <p>× 沒有提及排序次序 × 它判斷陣列是否未以降序排列</p>	1
(ii)	<p>ALG1 在迭代上的可讀性更高。</p> <p>× ALG1 的執行效率較高／ALG1 需要更少的儲存空間</p>	1
(iii)	<p>第 3 行 當 <code>i < 6 AND A = False</code> 執行</p>	1 1
(d) (i)	2	1
(ii)	<p><u>版本 1</u></p> <p>[Python 版本] <code>longest = 1 current = 1 for i in range(0, 6): if sales[i+1] > sales[i]: current = current + 1 if current > longest: longest = current else: current = 1</code></p> <p>[C++ 版本] <code>int longest = 1; int current = 1; for (int i = 0; i < 6; i = i+1) { if (sales[i + 1] > sales[i]) { current = current + 1; if (current > longest) { longest = current; } } else { current = 1; } } }</code></p> <p><u>版本 2</u></p>	5

[Python 版本]

```
longest = 1
current = 1

for i in range(0, 6):
    if sales[i+1] < sales[i]:
        current = 1
    else:
        current = current + 1
        if current > longest:
            longest = current
```

[C++ 版本]

```
int longest = 1;
int current = 1;
for (int i = 0; i < 6; ++i) {
    if (sales[i + 1] < sales[i]) {
        current = 1;
    } else {
        current = current + 1;
        if (current > longest) {
            longest = current;
        }
    }
}
```

以下任何 4 項：

- ① 正確的設循環（例如：`range(0, 6)` / `range(0, len(sales)-1)`）
- ① 正確的條件語句
- ① 遞增 `current`
- ① 重置 `current`
- ① 更新 `longest`

① 全部正確（除小錯誤外，語法正確）

小錯誤包括：

1. 在 Python 中，`for` 和 `if` 語句末尾漏掉冒號（`:`）
2. 在 C++ 或 Pascal 中，語句末尾漏掉分號（`;`）
3. 未考慮字母大小寫

以偽代碼形式表示的答案不給予任何分數

9. (a) $2^n \geq 310$
 $n \geq 15.8$
 \Rightarrow 需要 16 位元

2

其他答案:

$2^n \geq 3$
 $n \geq 1.58$
 每個燈泡需要 2 位元

$2 \times 10 = 20$ 位元，整個照明系統需要 20 位元

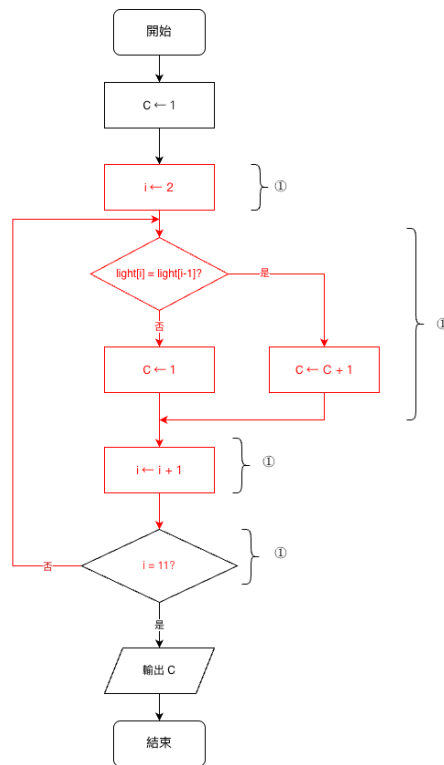
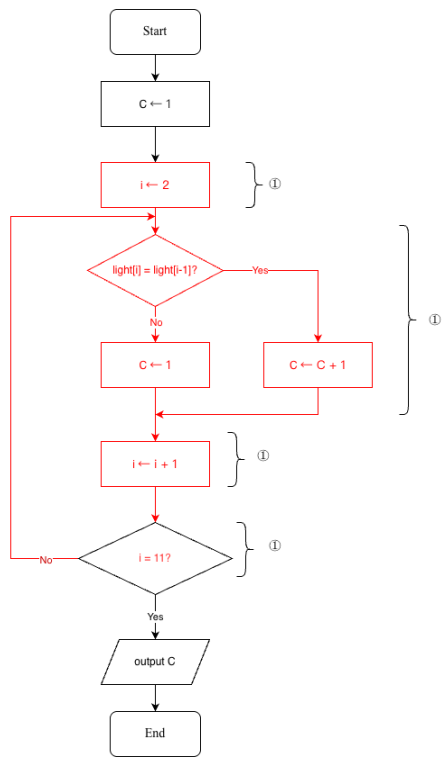
- ① 正確的公式
- ① 正確的答案

(b) Green/G

1

(c)

4



評分標準：

- ① 初始化 i ($i \leftarrow 2$)
- ① 正確的條件語句 (if)
- ① 遞增 i ($i \leftarrow i + 1$)
- ① 正確的判斷 ($i > 10 / i = 11$)

參考：偽代碼版本的後測循環

```

C ← 1
i ← 2
repeat
    if light[i-1] = light[i]
        C ← C + 1
    else
        C ← 1
        i ← i + 1
until i = 11

output i
    
```

```

C ← 1
i ← 2
重覆
    如果 light[i-1] = light[i]
        C ← C + 1
    否則
        C ← 1
        i ← i + 1
直至 i = 11

輸出 i
    
```

(d) (i) RGBRGRGRGR
RGRGRGRGRG
(或其他合理的答案)

1

(ii) 運行時錯誤

1, 1

分數

用於存取 `comp` 元素的索引超出了範圍

× 溢出錯誤

① 解釋正確但程式錯誤的名稱不正確

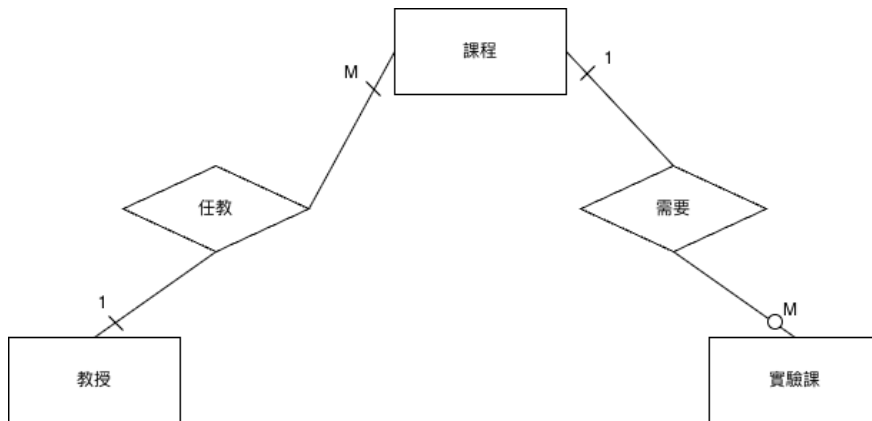
- (e) 如果 `idx >= 9` 則
離開此循環

1

1

1.

4



評分標準：

- ① 關係 (其中一個)
- ①,① 1:M
- ①,① 基數
- 5 個中的 4 個

2. (a) TicketID, SeatNum, ConcertID 1

(b) ConcertID 1

(c) ConcertID, ConcertName, Date, Performer, VenueID 1

(d) VenueID 1

(e) N/A 1

3. (a)

```
SELECT * FROM LOAN
WHERE StudentID = 'S202501' AND BOOKID = 'B0777';
```

2
① ①

(b)

```
CREATE INDEX idx_student_book
ON LOAN(STUDENTID, BOOKID);
```

2
①

- ① 複合索引
- ① 全部正確

4. (a) DELETE FROM CLASS 會移除數據庫表內的所有記錄 1
 DROP TABLE CLASS 會移除整個數據庫表

✗ 答案必須同時提及 DELETE 和 DROP

(b) UNION 會移除重複的列 1
 UNION ALL 會保留重複的列，因此會產生較多的列

✗ 答案必須同時提及 UNION 和 UNION ALL

- (c) S101
S103 2
- ① 任何一項記錄
① 全部正確
- 每個記錄之間用逗號分隔，扣1分，例如：s101, s103
5. (a) SELECT CustName
FROM CUSTOMER
WHERE YEAR(RegDate) = 2023;
① ① 2
- (b) SELECT ProdName
FROM PRODUCT P, SALES S
WHERE S.ProdID = P.ProdID and CustID = 'C1024';
① ① 2
- (c) SELECT ProdName
FROM PRODUCT
WHERE ProdID NOT IN (SELECT ProdID FROM SALES);
① 2
- ① 全部正確
- (d) SELECT Category, SUM(Quantity)
FROM PRODUCT P, SALES S
WHERE P.ProdID = S.ProdID ①
GROUP BY Category ①
HAVING SUM(Quantity) > 500; ① 3
- (e) 它會列出價格高於其所屬類別平均價格的產品名稱和價格。 2
- (f) **Sequence B** 2*
在 SALES 表中的 CustID 「C9002」無法參照 CUSTOMER 表中的 CustID，因此會出現參照完整性問題。
① 正確順序 (Sequence B)
① 正確解釋
6. 用於搜尋的輸入控制，例如：標籤按鈕、篩選器、價格滑桿
多媒體元素，包括：文字、音頻、視頻 3
7. (a) 優點：減少需人手分配的工作量／提高準確性 1
限制：某些裝置需要靜態 IP 位址，例如：檔案伺服器 1
- (b) 預設閘道／子網路遮罩 1×2
8. (a) 頻寬較高／穩定性較佳／覆蓋範圍較大 1
- (b) TCP：確保數據完整性 1
UDP：延遲較低 1

	分數
9. (a) (i) 用戶可能不知道可點擊的區域／流動裝置的優化較差／潛在的無障礙措施問題	1
(ii) 是。我們可以使用滑鼠點擊事件監聽器，並將用戶重新定向到另一個網頁。	1
(b) 購買伺服器：較為靈活（完全控制配置）	1
訂閱網頁寄存服務：設置較為簡單	1
10. (a) 標記使用了 HTML5。 （接受任何關於 HTML5 新功能的描述）	1
(b) 更新日期／作者／描述／建立日期	1×2
(c) 外部 CSS 可能包含額外的樣式，影響了 h1 的樣式。	1
(d) type="number"	1
(e) (i) 該檔案可以嵌入到同一網站的其他網頁中。 （模組化／數據獨立性）	1
(ii) (1) \$i + 1	1
(2) \$MOVIES[\$i]	1
(f) (i) <pre>const ADULT_PRICE = 100; const CONCESS_PRICE = 50; const totalPrice = ADULT_PRICE * numberOfAdult + CONCESS_PRICE * numberOfConcession;</pre>	2
① 一個錯誤	
② 全部正確	
(ii) <code>keyup / keydown / keypress / changed / blur / input</code> × <code>click / focus</code>	1
(iii) 這是一種只能在客戶端實現的用戶互動方式。	1
(g) (i) (1) <code>\$fields as \$field</code>	1
(2) <code>\$values . \$field . ", "</code>	1
(ii) 將用戶重新定向到網頁 <code>success.php</code> .	1
11. (a) <pre>isFull(S) return len(S) = 8</pre> ①	1
(b) 堆疊未滿時發生下溢 相關解決方案的描述	1 1
12. (a) (1) <code>j ← i + 1</code>	1
(2) <code>A[j] < A[j - 1]</code>	1
(3) <code>i >= N - 1 / i = N - 1</code>	1
(b) 快速排序／合併排序	1

- | | | |
|---------|--|---------|
| 13. (a) | 1 / SC[5, 0] | 分數
1 |
| (b) | ① 正確的百分比或正確的星號
① 全部正確

[Python 版本]
<pre>def star(per): cnt = 0 total_score = 0 N = 7 for i in range(N): if SC[i][per] != -1: total_score += SC[i][per] if SC[i][per] >= 5: cnt += 1 star = "" if cnt >= 3: star = "*" print("{}{}: {:.2f}".format(per, star, total_score/(N - 1)))</pre>
[C++ 版本]
<pre>void star(int per) { int cnt = 0; int total_score = 0; const int N = 7; char star = '\0'; for (int i = 0; i < N; i++) { if (SC[i][per] != -1) { total_score += SC[i][per]; } if (SC[i][per] >= 5) { cnt += 1; } if (cnt >= 3) { star = '*'; } std::cout << per << star << ": " << std::setprecision(3) << (float)total_score / (N - 1) << std::endl; } }</pre> | 1+1 |
| (c) (i) | 對於任何 i 與 j 的值， $\text{pair}(i, j) == \text{pair}(j, i)$ 。因此，當 $(i = b_i) > (j = b_j)$ 時， $\text{pair}(b_i, b_j)$ 必定等於 $\text{pair}(i, j)$ 。第 50 行的如果語句結果必定為 False。 | 1 |
| | (ii) 設 j 由 i+1 至 6 執行 / 設 j 由 i 至 6 執行 | 1 |
| 14. (a) | (1) <code>InsArr[i].dir = 'E'</code>
(2) <code>dy + InsArr[i].dis</code> | 1
1 |
| (b) | 超聲波感應器／紅外線感應器 | 1 |
| 15. (a) | 4.3 | 1 |
| (b) | ① 尋找最大值
① 累加
① 全部正確 | 3 |

[Python 版本]

```
max_S = S[0]
for i in range(1, C):
    if S[i] > max_S:
        max_S = S[i]

cnt = 0
for i in range(0, C):
    if S[i] == max_S:
        cnt = cnt + 1

print(cnt)
```

[C++ 版本]

```
double max_S = S[0];

for (int i = 1; i < C; i++) {
    if (S[i] > max_S) {
        max_S = S[i];
    }
}

int cnt = 0;

for (int i = 0; i < C; i++) {
    if (S[i] == max_S) {
        cnt += 1;
    }
}

std::cout << cnt << std::endl;
```

- (c) (i) ① 迭代 3
 ① 循環
 ① 全部正確

[偽代碼]

```
isExist ← False
設 j 由 0 至 size - 1 執行
    如果 arr[i] = e1 則
        isExist ← True
傳回 isExist
```

- (ii) 當 N 中有重複元素時，NX[i] 可能會超出索引。 1
 第 70 行：NX[size - 1] ← N[i] 1

- (iii) 除了使用 (c)(i) 中的函數，我們可以檢查 N[i - 1] 中是否存在相同的值。 2*

- (d) ① 循環 1+1
 ① 全部正確

[Python 版本]

```
def getAverage(T):
    cnt = 0
    total = 0
```

```
for i in range(C):  
    if N[i] == T:  
        cnt = cnt + 1  
        total = total + S[i]  
return total/cnt
```

[C++ 版本]

```
double getAverage(std::string T) {  
    int cnt = 0;  
    double total = 0;  
  
    for (int i = 0; i < C; i++) {  
        if (S[i] == T) {  
            cnt += 1;  
            total += N[i];  
        }  
    }  
  
    return total / cnt;  
}
```

- (e) 優點：插入／刪除時間較短
缺點：只支援順序存取／需要更多記憶體空間

1
1