

香港考試及評核局  
2016年香港中學文憑考試

## 資訊及通訊科技

### 試卷二 (D)

#### 軟件開發

#### 試題答題簿

本試卷必須用中文作答  
一小時三十分鐘完卷  
(上午十一時十五分至下午十二時四十五分)

#### 考生須知

- (一) 宣布開考後，考生須首先在第1頁之適當位置填寫考生編號，並在第1、3、5及7頁之適當位置貼上電腦條碼。
- (二) 在合適的方格選取所採用的程式編寫語言。若選取超過一個方格或不選取任何方格，將不獲給分。
- (三) 本試卷全部試題均須回答。答案須寫在本試題答題簿中預留的空位內。不可在各頁邊界以外位置書寫。寫於邊界以外的答案，將不予評閱。
- (四) 如有需要，可要求派發補充答題紙。每一紙張均須填寫考生編號、填畫試題編號方格、貼上電腦條碼，並用繩縛於簿內。
- (五) 試場主任宣布停筆後，考生不會獲得額外時間貼上電腦條碼及填畫試題編號方格。

請在此貼上電腦條碼

考生編號									
採用的程式 編寫語言 (請選一項)	Pascal	<input type="checkbox"/>							
	C	<input type="checkbox"/>							
	Visual Basic	<input type="checkbox"/>							
	Java	<input type="checkbox"/>							



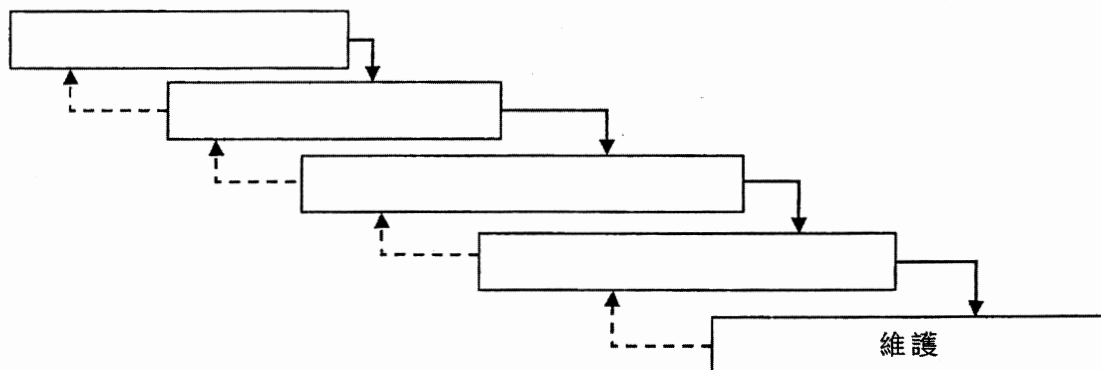
本試卷全部試題均須回答。

1. 使用鍵盤打字時可能會錯誤地倒轉了兩個相鄰的字符，例如把「sequence」打成為「sequecne」。志聰打算開發一個打字修正系統 (TCS) 去改正這類打字錯誤。

(a) CmpRStr 是 TCS 的核心子程式。志聰將會用「瀑布模式」開發 CmpRStr。

(i) 填上以下「瀑布模式」中的階段：

- (1) 整合
- (2) 設計
- (3) 要求
- (4) 實施



(ii) 選擇算法應在哪個階段完成？ \_\_\_\_\_

(iii) CmpRStr 在哪個階段可以獨立地執行？ \_\_\_\_\_

(iv) 志聰曾經考慮使用應用系統迅速發展法 (RAD)。試舉出 RAD 的一個限制。

\_\_\_\_\_  
\_\_\_\_\_

(7 分)

- (b) 編譯 CmpRStr 之後會涉及連接程式和載入程式。試描述連接程式和載入程式的主要功能。

連接程式： \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

載入程式： \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

(4 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

CmpRStr(S, T) 是 TCS 的核心子程式，其偽代碼如下所示。S 和 T 是陣列，儲存了兩個字串。假設 S 和 T 中的首個字符的索引是 0。

```

CmpRStr(S, T)
  len ← S 的字長
  count ← 0
  如果 len = T 的字長 則
    j ← 0
    當 j < len-1 便執行
      如果 S[j] ≠ T[j] 則
        如果 (S[j] = T[j+1]) 與 (S[j+1] = T[j]) 則
          count ← count + 1
          j ← j + 1
        否則
          count ← -1
          j ← len
      j ← j + 1
    如果 j = len-1 則
      如果 S[len-1] ≠ T[len-1] 則
        count ← -1
    傳回 count
  否則 傳回 -1
  
```

(c) 就以下 S 和 T 的值，寫出 CmpRStr 傳回的值。

S	T	CmpRStr 傳回的值
banana	banna	
banana	canana	
banana	abanan	
banana	banank	

(4 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

2. 陳先生計畫為某餐廳開發一個流動應用程式。顧客能利用該流動應用程式預訂餐枱，當他們的座位準備好時便會收到一個通知。陳先生使用以下變量和函數去處理輪候顧客的預訂：

變量 / 函數	描述
A	索引由 0 至 n-1 的陣列
start	儲存首位輪候顧客在 A 內位置的變量
next	儲存新顧客在 A 內位置的變量
addG(name)	一個子程式；假如輪候顧客的數目少於 n，會加入新顧客姓名 name 到 A 內
removeG	一個子程式；假如 A 不是空的，會傳回 A 內首位輪候顧客的名字，及將此名字從 A 中移除

例如：假設  $n = 8$ ，而且沒有顧客正在輪候。

start = 0, next = 0

i	0	1	2	3	4	5	6	7
A 的第 i 個項目								

順序調用 addG(Amy)、addG(Ben)、addG(Candy)、addG(Den)、addG(Eda)、RemoveG()、RemoveG()、addG(Eric)、addG(Fred)、addG(Gail) 和 addG(Hank) 後，

start = 2, next = 1

i	0	1	2	3	4	5	6	7
A 的第 i 個項目	Hank		Candy	Den	Eda	Eric	Fred	Gail

- (a) (i) 餐廳的座位分配是先到先得的。A 是屬於哪類數據結構？

- (ii) 假設  $n = 8$ ，

start = 4, next = 1

i	0	1	2	3	4	5	6	7
A 的第 i 個項目	Ken				Mike	Belle	Joe	June

在下列填寫順序調用 addG(Joan)、RemoveG() 和 addG(Lily) 後的值。

start = , next =

i	0	1	2	3	4	5	6	7
A 的第 i 個項目								

(4 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

(b) 完成以下 addG(name) 和 RemoveG() 的偽代碼：

```

addG(name)
    如果 start = ((next+1)/n) 的餘數 則
        退出及附以提示信息「陣列已滿」
    否則
        A[  ] ← name
        next ← 
    結束
    
```

```

RemoveG()
    如果 next =  則
        退出及附以提示信息「沒有顧客」
    否則
        start ← ((start + 1)/n) 的餘數
        i ← 
        temp ← A[i]
        A[i] ← 空值
        傳回 temp
    結束
    
```

(4 分)

(c) 陳先生打算展示當前在 A 內預訂餐枱的總數目。寫出以下每種情況當前預訂餐枱的總數目的算式：

情況 1: next ≥ start

情況 2: next < start

(3 分)

陳先生使用另一個陣列 S 去儲存個別輪候顧客所要求的座位數目。在以下例子中「Peter」要求了一張 8 人餐桌。

start = 5, next = 3

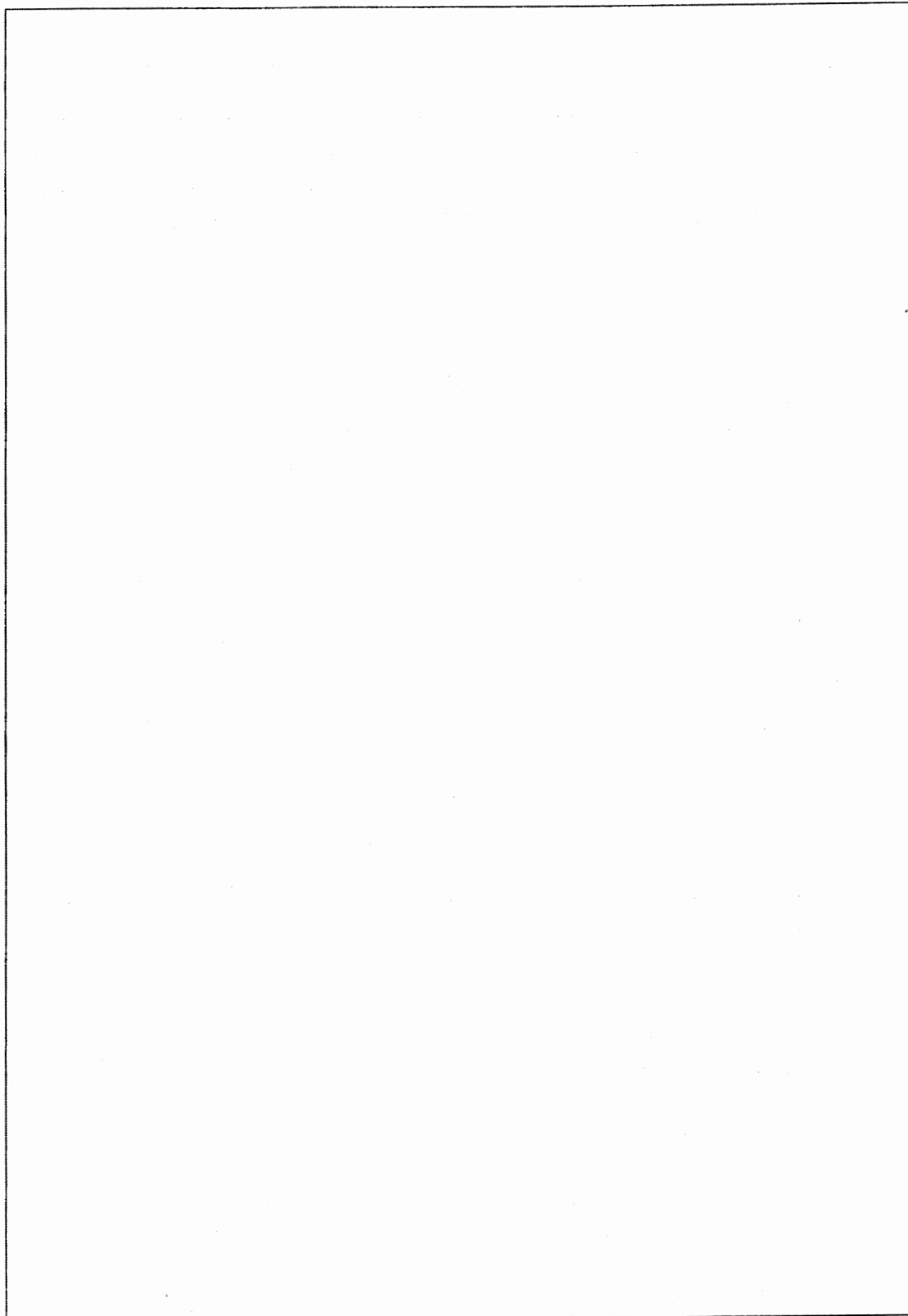
i	0	1	2	3	4	5	6	7
A 的第 i 個項目	Tom	Betty	Ken			Peter	John	Mary

i	0	1	2	3	4	5	6	7
S 的第 i 個項目	2	4	2			8	4	2



(d) 對於任何  $n$  值，寫出偽代碼來顯示當前預訂兩個座位餐枱的總數。



(5 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

3. 志偉開發了一種填色方法去塗黑 4×4 像素的圖像。圖像的所有像素都有一個由數字 1 至 16 的標籤。另外使用四個額外數字 17、18、19 和 20，分別代表像素群組 1 至 4 像素、5 至 8 像素、9 至 12 像素和 13 至 16 像素。就一個圖像而言，志偉會使用一最短且按升序排列的數字，以記錄要塗黑的像素。

例如，以下圖像按「4, 13, 15, 16, 18」來塗黑：

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

- (a) 採用志偉的填色方法。

- (i) 以下圖像按什麼序列塗黑？

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

- (ii) 按「1, 5, 6, 19」將以下圖像塗黑：

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

- (iii) 按一最長可行的序列，將以下圖像塗黑：

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(4 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

志偉計畫利用以下變量編寫一個子程式，從一個序列去重組圖像。

變量	描述
A	儲存序列的整數陣列
N	儲存序列長度的整數變量
P	索引由 1 至 16 的字符陣列，儲存像素的顏色 (「B」和「W」分別代表黑色和白色。)

例子如下所示：

已知該圖像的 A 和 N 內容是

	i	1	2	3	4	5	
A 的第 i 個項目		4	13	15	16	18	N 5

此子程式將數據解碼，並將結果儲存在 P 內，以代表其圖像，如下展示：

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

	i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P 的第 i 個項目		W	W	W	B	B	B	B	B	W	W	W	W	B	W	B	B

(b) 假設初始時 P 的內容是空的。寫出此子程式的偽代碼。

(5 分)

寫於邊界以外的答案，將不予評閱。



志偉改良他的填色方法，因應數字的序列，計算像素會被引用多少次。如果某像素只被引用一次，它便是黑色；否則，像素是白色。例如，以下圖像以「4, 14, 20」來塗黑。其中數字 4、13、15 和 16 標籤的像素被引用了一次，所以是黑色。而數字 14 標籤的像素被 14 和 20 引用了兩次，所以是白色。

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(c) 採用志偉改良後的方法。

(i) 以下圖像按什麼序列塗黑？

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(ii) 按「2, 5, 17, 19」將以下圖像塗黑：

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(3 分)

(d) 比較志偉原本的方法和改良後的方法。

(i) 試舉出一個有 7 個黑色像素的圖像，採用這兩個方法所得出的序列長度均是相同的。

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(ii) 試舉出一個圖像，採用這兩個方法所得出的序列長度之差是最大的。

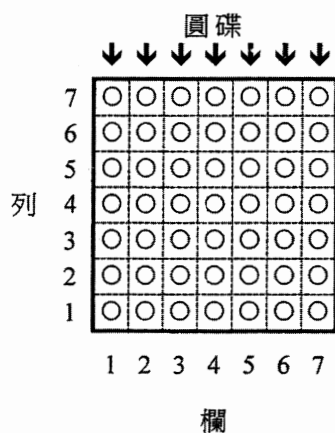
1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

(3 分)

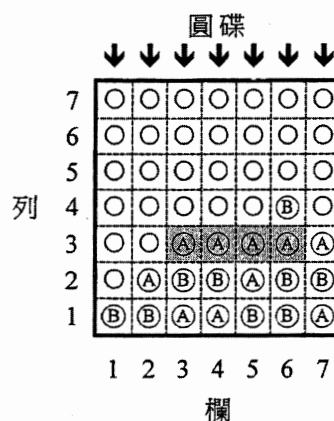
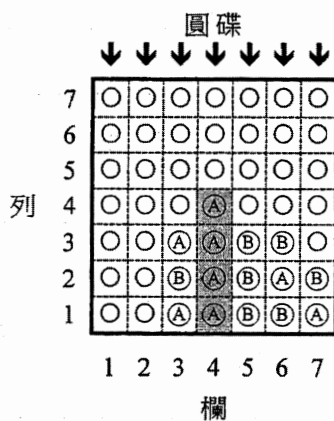
寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

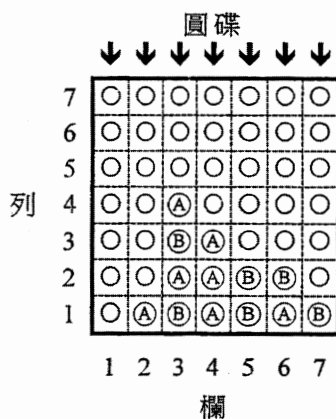
4. 李先生計畫為「直 4 遊戲」編程，這是一個以 7×7 遊戲板進行的雙人遊戲，如下所示。玩家每次輪流由上面投入一個圓碟到一個欄中。圓碟會被放置在欄中最低但並未被佔用的圓圈內。



當玩家有四個圓碟已連續地放置在一列或一欄內便為勝出。以下兩個例子中，使用圓碟「A」的玩家皆勝出。



「直 4 遊戲」程式的偽代碼中，李先生使用了兩個全程陣列 BD 和 topC。此遊戲板的例子和一些樣本數值展示如下：



寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

變量	描述	例子	數值
BD	儲存 7×7 遊戲板上圓圈的內容的全程二維字符陣列 (首和次索引分別代表欄號和列號)	BD[2,1] BD[3,1]	A B
topC	一個全程整數陣列，儲存欄中最低但並未被佔用的圓圈的列號 (索引代表欄號)	topC[3]	5

李先生寫了以下子程式 putDisc 的偽代碼，將一圓碟放置在遊戲板內。

```

col ← 將要放置的欄
player ← 玩家的圓碟
如果 (1 ≤ col ≤ 7) 與 (topC[col] ≤ 7) 則
    BD[col, topC[col]] ← player
    topC[col] ← topC[col] + 1
    傳回 true
否則
    傳回 false

```

(a) (i) 在「如果」語句中，條件「 $1 \leq \text{col} \leq 7$ 」有什麼目的？

---



---

(ii) 在「如果」語句中，條件「 $\text{topC}[\text{col}] \leq 7$ 」有什麼目的？

---



---

(iii) 若 putDisc 傳回的值是 false，「直 4 遊戲」程式應會接着做什麼？

---



---

(3 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

李先生使用子程式 checkCol 和 checkRow 分別檢查是否有一個玩家已有四個圓碟連續地放置在一欄或一行中，這項檢查會在執行 putDisc 後進行。若有玩家勝出，它們會傳回 true，否則會傳回 false。除 BD 和 topC 外，李先生在這些子程式中使用以下變量：

變量	描述
col	儲存剛投入圓碟的欄號的整數變量
player	儲存圓碟標籤的字符變量，其標籤代表玩家的身分
connected	計算在檢查期間相連圓碟數目的整數變量
i	暫存整數變量

(b) 完成以下 checkCol 的偽代碼：

```

col ← 剛投入圓碟的欄號
player ← 玩家的圓碟
connected ← 1

i ← topC[ ] - 2
當 (i ≥ 1) 與 (BD[ , ] = player) 便執行
    connected ← connected + 1
    i ← i - 1
如果 (connected = )
則    傳回 true
否則  傳回 false
    
```

(4 分)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

- (c) 以 Pascal、C、Visual Basic 或 Java 編寫 checkRow，其首行如下所示。考生可以直接使用 (a) 和 (b) 部的變量，如有需要，也可定義其他變量。在 C 版本中，checkRow 傳回 1 和 0 分別代表布爾值「真」和「假」。

<b>Pascal 版本</b>	Function checkRow(col:integer; player:char):boolean
<b>C 版本</b>	int checkRow(int col, char player)
<b>Visual Basic 版本</b>	Function checkRow(col As Integer, player As char) As Boolean
<b>Java 版本</b>	boolean checkRow(int col, char player)

寫於邊界以外的答案，將不予評閱。

寫於邊界以外的答案，將不予評閱。

試卷完

(7 分)

寫於邊界以外的答案，將不予評閱。