

香港考試及評核局
2013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

一小時三十分完卷

(上午十一時十五分至下午十二時四十五分)

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"/> | | | |

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。



本試卷全部試題均須回答。

1. (a) 下列算法 ALG1 處理指數 1 至 n 的整數陣列 x。

ALG1

步驟 1：設 i 由 1 至 n-1 執行步驟 2 至 6
步驟 2： 設 j 由 1 至 n-1 執行步驟 3 至 6
步驟 3： 如果 $x[j] > x[j+1]$ ，則執行步驟 4 至 6
步驟 4： $k \leftarrow x[j] + x[j+1]$
步驟 5： $x[j] \leftarrow k - x[j]$
步驟 6： $x[j+1] \leftarrow k - x[j]$

- (i) 假設 $n = 6$ 。x 的初始內容如下所示。空運行 ALG1。

| x[1] | x[2] | x[3] | x[4] | x[5] | x[6] |
|------|------|------|------|------|------|
| 5 | 6 | 2 | 3 | 1 | 4 |

- (1) 填上步驟 1 的循環第一遍和第二遍迭代後 x 的內容。

第一遍迭代後

| x[1] | x[2] | x[3] | x[4] | x[5] | x[6] |
|------|------|------|------|------|------|
| | | | | | |

第二遍迭代後

| x[1] | x[2] | x[3] | x[4] | x[5] | x[6] |
|------|------|------|------|------|------|
| | | | | | |

- (2) 填上 x 的最終內容。

| x[1] | x[2] | x[3] | x[4] | x[5] | x[6] |
|------|------|------|------|------|------|
| | | | | | |

- (3) 步驟 3 內的語句會被執行多少次？ _____

- (ii) 有一建議將步驟 2 修訂為「設 j 由 1 至 $n-i$ 執行步驟 3 至 6」可改善此算法。你同意嗎？試加說明。

- (iii) 假設只需計算及輸出 x 內第二最大值。

- (1) 執行 ALG1 後應輸出哪項？ $x[\underline{\hspace{2cm}}]$

- (2) 修改步驟 1，令執行步驟 3 的次數減到最少。

設 i 由 _____ 至 _____ 執行步驟 2 至 6

(9 分)

寫於邊界以外的答案，將不予以評閱。

請在此貼上電腦條碼

- (b) 下列算法 ALG2 處理指數 1 至 n 的整數陣列 P 和 Q 。在這些陣列內的整數均以倒序儲存。 X 是指數 1 至 $(n+n)$ 的整數陣列。

ALG2

步驟 1: $i \leftarrow n$; $j \leftarrow n$; $k \leftarrow 1$
步驟 2: 當 $k \leq (n+n)$ 執行步驟 3 至 10
步驟 3: 如果 $j=0$
步驟 4: 則 $X[k] \leftarrow P[i]$ 和 $i \leftarrow i - 1$
步驟 5: 否則 如果 $i=0$
步驟 6: 則 $X[k] \leftarrow Q[j]$ 和 $j \leftarrow j - 1$
步驟 7: 否則 如果 $P[i] < Q[j]$
步驟 8: 則 $X[k] \leftarrow P[i]$ 和 $i \leftarrow i - 1$
步驟 9: 否則 $X[k] \leftarrow Q[j]$ 和 $j \leftarrow j - 1$
步驟 10: $k \leftarrow k + 1$

假設 $n = 6$ 。 P 和 Q 的初始內容如下所示。空運行 ALG2。

| $P[1]$ | $P[2]$ | $P[3]$ | $P[4]$ | $P[5]$ | $P[6]$ |
|--------|--------|--------|--------|--------|--------|
| 12 | 10 | 9 | 6 | 2 | 1 |

| $Q[1]$ | $Q[2]$ | $Q[3]$ | $Q[4]$ | $Q[5]$ | $Q[6]$ |
|--------|--------|--------|--------|--------|--------|
| 11 | 8 | 7 | 5 | 4 | 3 |

- (i) $X[1]$ 和 $X[12]$ 的最終內容是什麼？

| | |
|--------|---------|
| $X[1]$ | $X[12]$ |
| | |

- (ii) 步驟 3 內的語句會被執行多少次？_____

- (iii) 以單一的「如果-則-否則」語句簡化步驟 3 至 9。

寫於邊界以外的答案，將不予以評閱。

寫於邊界以外的答案，將不予以評閱。

2. 算法 CAL 處理儲存在陣列 N 內算術算式的操作。N 內有整數及符號「(」、「)」、「+」和「-」。CAL 採用下列堆疊 S 的操作：

push(K, S) 將元素 K 置於 S 的頂部
pop(S) 移除及傳回 S 頂部的元素

CAL

步驟 1： 初始化 S 為空的
步驟 2： $i \leftarrow 1$
步驟 3： 當 $N[i]$ 不是空時便執行步驟 4 至 12
步驟 4： 如果 $N[i] = ')'$ 則執行步驟 5 至 10
步驟 5： $A \leftarrow pop(S)$
步驟 6： $B \leftarrow pop(S)$
步驟 7： $C \leftarrow pop(S)$
步驟 8： $temp \leftarrow pop(S)$
步驟 9： 如果 $B = '+'$ 則 $push(C + A, S)$
步驟 10： 否則 $push(C - A, S)$
步驟 11： 否則 $push(N[i], S)$
步驟 12： $i \leftarrow i + 1$
步驟 13： 輸出 $pop(S)$

- (a) CAL 是將要在一電腦系統內實行的算法，而有關程式將會在此系統內多次被執行。這樣應使用什麼翻譯器？試加說明。

(2 分)

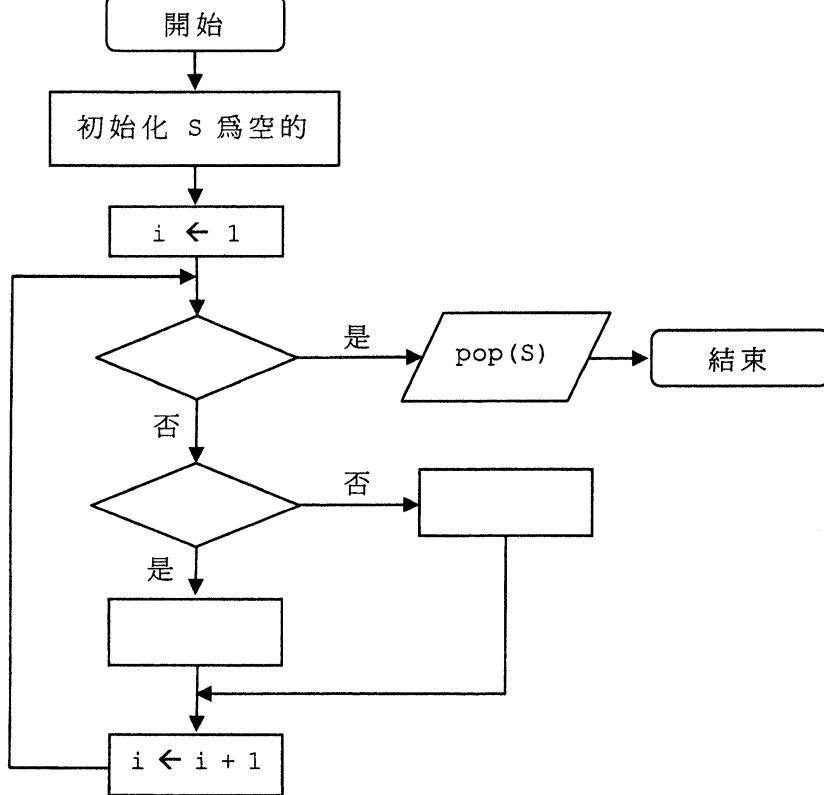
- (b) 試從以下 ① 至 ⑥ 的語句中揀選四句來完成下列 CAL 的流程圖。

- ① 符號是「+」？
- ② 插入符號至 S 內
- ③ $N[i]$ 是空的
- ④ 完成運算及插入此結果至 S 內
- ⑤ $N[i] = ')'$ ？
- ⑥ 移除 S 的元素

寫於邊界以外的答案，將不予以評閱。

寫於邊界以外的答案，將不予以評閱。

請在此貼上電腦條碼



(4 分)

寫於邊界以外的答案，將不予以評閱。

寫於邊界以外的答案，將不予以評閱。

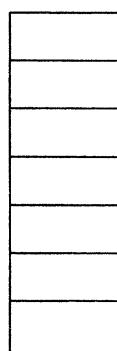
(c) 已知 N 的初始內容如下。

| N[1] | N[2] | N[3] | N[4] | N[5] | N[6] | N[7] | N[8] | N[9] | N[10] |
|------|------|------|------|------|------|------|------|------|-------|
| (| (| 2 | + | 3 |) | - | 4 |) | |

空運行 CAL，並在下方寫出 s 的內容。

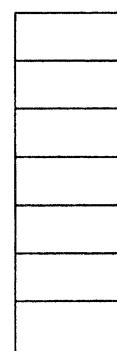
在步驟 12 中，

當 i 由 5 轉變為 6 時



← s 的底部

當 i 由 6 轉變為 7 時



← s 的底部

(4 分)

寫於邊界以外的答案，將不予以評閱。

(d) 假設 s 以陣列實行，而 N 的內容是有效的及 N 的大小為 10。 s 的大小最小是什麼？

(2 分)

(e) 下列三組測試用例將會用來測試 CAL。

A組

| 測試用例 | N[1] | N[2] | N[3] | N[4] | N[5] | N[6] | N[7] | N[8] | N[9] |
|------|------|------|------|------|------|------|------|------|------|
| A1 | (| 1 | + | 2 |) | | | | |
| A2 | (| 1 | - | 2 |) | | | | |
| A3 | (| (| 1 | - | 2 |) | + | 3 |) |

B組

| 測試用例 | N[1] | N[2] | N[3] | N[4] | N[5] | N[6] | N[7] | N[8] | N[9] |
|------|------|------|------|------|------|------|------|------|------|
| B1 | 1 | + | 2 |) | | | | | |
| B2 | (| 1 | + | 2 |) |) | | | |
| B3 | (| 1 | + | 2 |) | + | 3 | | |

C組

| 測試用例 | N[1] | N[2] | N[3] | N[4] | N[5] | N[6] | N[7] | N[8] | N[9] |
|------|------|------|------|------|------|------|------|------|------|
| C1 | | | | | | | | | |

C2

寫出這三組測試用例的不同用途。

A組：_____

B組：_____

C組：_____

(3 分)

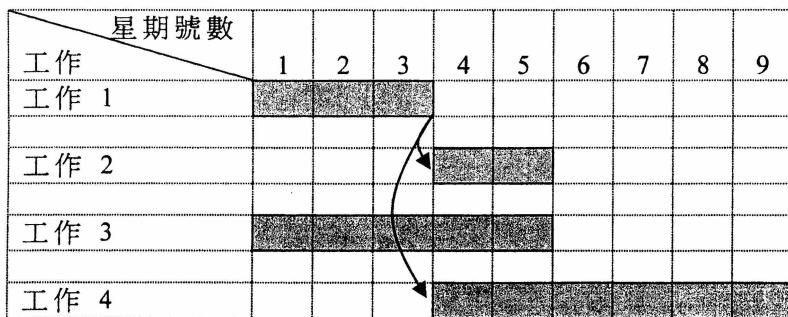
寫於邊界以外的答案，將不予以評閱。

寫於邊界以外的答案，將不予以評閱。

請在此貼上電腦條碼

3. 李先生從事一個項目，開發一個網上拍賣系統。用戶可透此系統提交拍賣物件資料來建立一項拍賣，或出價購買拍賣物件。

- (a) 李先生為這個項目建構一甘特圖，如下所示。

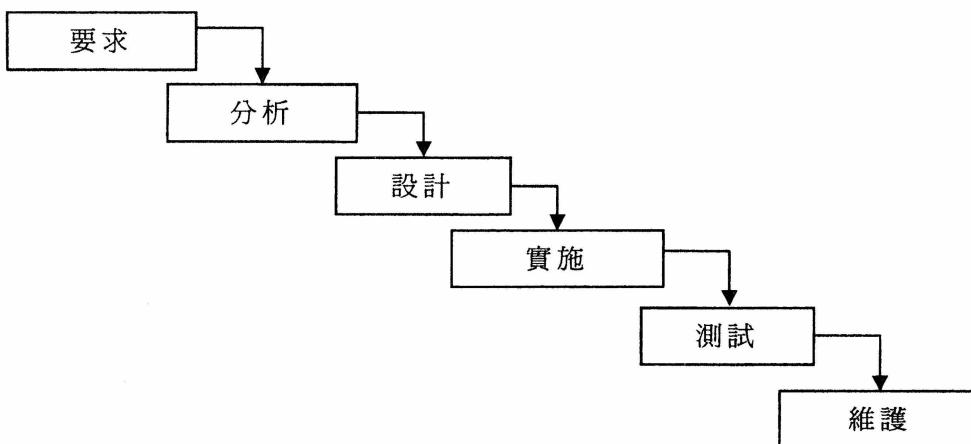


(i) 李先生計畫需多少個星期才可完成這個項目？_____

(ii) 這個甘特圖內工作 1 和工作 2 之間有什麼關係？

(2 分)

李先生在系統開發時採用下列瀑布模式。



(b) 在測試階段內，李先生發現此系統未能通過用戶驗收測試。錯誤可能會在哪個(些)階段發生？他可如何查明？

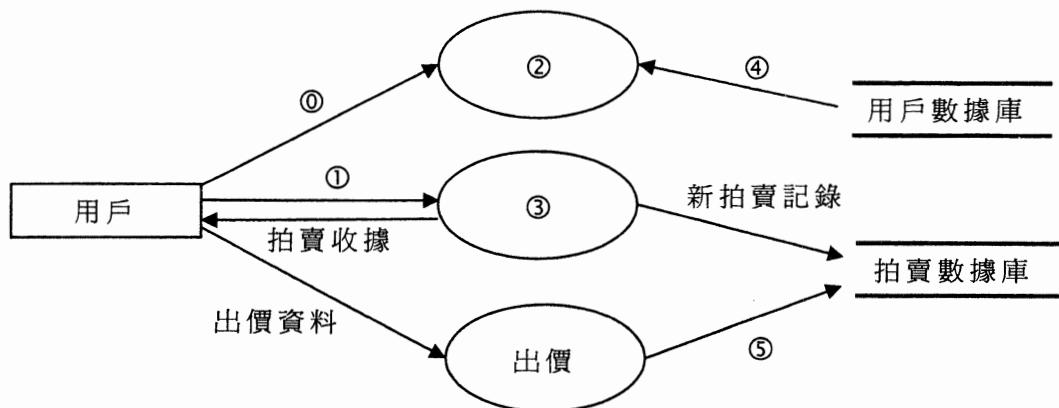
(3 分)

寫於邊界以外的答案，將不予評閱。

(c) 李先生為展示該網上拍賣系統內數據的流動而建構了一個數據流程圖。

(i) 根據上述瀑布模式，此數據流程圖應在哪一階段創建？

(ii) 完成下列網上拍賣系統的數據流程圖。寫出各項目的數字。



| 項目 | 數字 |
|---------|----|
| 用戶名稱／密碼 | ① |
| 拍賣物件資料 | |
| 出價記錄 | |
| 創建拍賣 | |
| 鑑定 | |
| 用戶資料 | |

(6 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。

- (d) 已知 myRAND 是一個子程式，可從 1 至 1000 之間隨機選取一個整數，包括首尾兩數。李先生設計下列算法 R1 來隨機選取一個拍賣項目。

R1
步驟 1: $n \leftarrow$ 拍賣項目數目
步驟 2: $i \leftarrow (\text{myRAND}() \div n)$ 的餘數 + 1
步驟 3: 傳回第 i 個物件

- (i) 以 Pascal、C、Visual Basic 或 Java 編寫 myRAND，使每當執行 myRAND 時，電腦均會傳回不同的隨機數字。

- (ii) 李先生發現有些拍賣物件可能永不會被 R1 選取。拍賣物件的總數會是多少？

- (iii) 李先生發現有些拍賣物件可能較其他物件更常被 R1 選取。拍賣物件的總數會是多少？

(5 分)

4. 小麗利用一軟件包來儲存一些 4×4 像素的黑白圖像為文字檔。此軟件包以下列方法 1 和方法 2 將這些圖像儲存，在這兩個方法內均以「1」和「0」分別代表黑色像素和白色像素。

方法 1：圖像以包含「1」和「0」的 4×4 字符的文字檔來儲存。圖像內每一像素均以此檔案內對應的字符表示。

方法 2：圖像從最頂一行，由左至右掃描。文字檔內以多組雙數字 (P, Q) 表示像素， P 是數字「1」或「0」（黑色／白色像素）， Q 是連續出現這個數字的總數。

例子 1 展示此軟件包如何儲存圖像。

| | 圖像 | 方法 1 | 方法 2 | | | | | | | | | | | | | | | | | | | | | | |
|------|----|--|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 例子 1 | | <table border="1"><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | <table border="1"><tr><td>1</td><td>3</td></tr><tr><td>0</td><td>5</td></tr><tr><td>1</td><td>8</td></tr></table> | 1 | 3 | 0 | 5 | 1 | 8 |
| 1 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 3 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 5 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 8 | | | | | | | | | | | | | | | | | | | | | | | | |

(a) 某圖像以方法 2 儲存，如下所示。請在右方填上此圖像的黑色像素。

| 方法 2 | 圖像 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| <table border="1"><tr><td>0</td><td>4</td></tr><tr><td>1</td><td>2</td></tr><tr><td>0</td><td>2</td></tr><tr><td>1</td><td>2</td></tr><tr><td>0</td><td>6</td></tr></table> | 0 | 4 | 1 | 2 | 0 | 2 | 1 | 2 | 0 | 6 | <table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table> | | | | | | | | | | | | | | | | | | | | |
| 0 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

(2 分)

(b) (i) 試從檔案大小的角度來看，描述圖像以方法 2 儲存的一個最佳例證和一個最差例證。

最佳例證：

最差例證：

(ii) 除檔案大小外，舉出方法 1 勝於方法 2 的一個優點。

(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。

(c) 某圖像以方法 1 儲存在一個文字檔，而一全程雙陣列 BD 儲存了此文字檔內的數據，BD 內索引(1, 1) 和 (4, 4) 的陣列元素分別儲存了左上角和右下角的數字。

小麗打算利用 BD 編寫子程式 ENC，把此圖像以方法 2 儲存為文字檔，並採用下列變量來儲存此文字檔內的數據。

| 變量 | 描述 |
|----|---------------------------|
| P | 儲存每組 (P, Q) 內首個數值的全程整數陣列 |
| Q | 儲存每組 (P, Q) 內第二個數值的全程整數陣列 |

以例子 1 為例，

| Pascal / C / Java 版本 | Visual Basic 版本 |
|----------------------|--------------------|
| P[1] = 1, Q[1] = 3 | P(1) = 1, Q(1) = 3 |
| P[2] = 0, Q[2] = 5 | P(2) = 0, Q(2) = 5 |
| P[3] = 1, Q[3] = 8 | P(3) = 1, Q(3) = 8 |

(i) 完成 ENC。

[Pascal 版本]

```

procedure ENC;
var
  i, j, k, current : integer;
begin
  k := 1;
  P[1] := BD[1,1];
  Q[1] := [REDACTED];
  current := [REDACTED];
  for i := 1 to 4 do
    for j := 1 to 4 do
      if (BD[i,j] = [REDACTED]) then
        Q[k] := Q[k] + [REDACTED]
      else begin
        k := k + 1;
        P[k] := BD[i,j];
        Q[k] := [REDACTED];
        current := BD[i,j];
      end;
end;

```

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 版本]

```
void ENC() {  
    int i, j, k, current;  
    k = 1;  
    P[1] = BD[1][1];  
    Q[1] = [ ];  
    current = [ ];  
    for (i=1; i<=4; i++)  
        for (j=1; j<=4; j++)  
            if (BD[i][j] == [ ])  
                Q[k] = Q[k] + [ ];  
            else {  
                k++;  
                P[k] = BD[i][j];  
                Q[k] = [ ];  
                current = BD[i][j];  
            }  
}
```

[Visual Basic 版本]

```
Sub ENC()  
    Dim i, j, k, current As Integer  
    k = 1  
    P(1) = BD(1,1)  
    Q(1) = [ ]  
    current = [ ]  
    For i = 1 to 4  
        For j = 1 to 4  
            If BD(i,j) = [ ] Then  
                Q(k) = Q(k) + [ ]  
            else  
                k = k + 1  
                P(k) = BD(i,j)  
                Q(k) = [ ]  
                current = BD(i,j)  
            End If  
        Next j  
    Next i  
End Sub
```

寫於邊界以外的答案，將不予評閱。

[Java 版本]

```
void ENC() {  
    int i, j, k, current;  
    k = 1;  
    P[1] = BD[1][1];  
    Q[1] = [ ] ;  
    current = [ ] ;  
    for (i=1; i<=4; i++)  
        for (j=1; j<=4; j++)  
            if (BD[i][j] == [ ])  
                Q[k] = Q[k] + [ ] ;  
            else {  
                k++;  
                P[k] = BD[i][j];  
                Q[k] = [ ] ;  
                current = BD[i][j];  
            }  
}
```

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。

- (ii) 小麗希望減省 ENC 所佔用的記憶體，她認為可重新編寫它，並只需使用 P 首個元素，換句話說，P 內的其他元素並不需要。你同意嗎？試簡略說明。

(7 分)

- (d) 小麗考慮採用物件導向語言或過程語言來編寫此子程式。舉出每種程式編寫語言對這項工作的一個優點。

物件導向語言：_____

過程語言：_____

(2 分)

試卷完

寫於邊界以外的答案，將不予評閱。