

卷二 (D)

分

1. (a) (i) (1)

X[1]	X[2]	X[3]	X[4]	X[5]	X[6]
0	1	1	1	0	1

(-1 一個錯誤)

2

(2) 5

1

(ii) N 介乎 0 至 63 之間。

1

當 X 的所有 6 個元素填滿 0，則是轉換了 N 的最小值至二進制，即是 0。

當 X 的所有 6 個元素填滿 1，則是轉換了 N 的最大值至二進制，即是 63。

1

(iii) 算法是用來轉換十進制數值 N 至二進制的數字。

1

(b) (i)

Y[1]	Y[2]	Y[3]	Y[4]	Y[5]	Y[6]
1	1	0	0	0	0

(-1 一個錯誤)

2

(ii) 6

1

(iii) N = 32

1

當 N = 32，算法 ALG2 執行步驟 6 一遍後，N = 0，而 While 循環執行第一遍後即終止。

1

當 N = 32，算法 ALG1 執行步驟 6 共六遍。

1

其他可行的 N 值：

N	二進制	執行次數(ALG1)	執行次數(ALG2)
8	001000	4	3
16	010000	5	2
20	010100	5	4
24	011000	5	3
28	011100	5	4
32	100000	6	1
34	100010	6	5
36	100100	6	4
38	100110	6	5
40	101000	6	3
42	101010	6	5
44	101100	6	4
46	101110	6	5
48	110000	6	2
50	110010	6	5
52	110100	6	4
54	110110	6	5
56	111000	6	3
58	111010	6	5
60	111100	6	4
62	111110	6	5

(c) (i) 「ICT bytecode」可在不同平台上執行 (或「ICT bytecode」具可移植性)。

1

(ii) (1) 須要生成可執行的機械碼檔案。

1

(2) 程式執行得較快。／

1

因為沒有使用 ICT-VM，需要較少資源運行程式。

	分
2. (a) (i) 16	1
X = 4	1
(ii) (1) 第 5 至第 19 天的到達時間會被刪除掉。(損失數據)／ 堆疊指示標指向錯誤的元素來儲存新的到達時間。(錯誤指示標)	1 1
(2) 複製堆疊至另一臨時的堆疊，並從該臨時堆疊讀取數據。 (儲存數據處理)	1
(iii) 堆疊會產生上溢錯誤。	1
(b) (i) 員工編號	1
(ii) 一個指示員工編號是否有效的標記。	2
(ⓐ 標記)	
(ⓑ 目的)	
(c) (i) (1) 單元測試	1
(2) 用戶驗收測試	1
(3) 驗收測試：確保系統符合該公司的要求。	1
系統測試：評估整個系統在統合個別模組後是否符合規格的需求。	1
單元測試：確保每個模組都會按已定義的規格執行其功能。	1
(ii) (1) 若新系統有任何問題，整個系統均受到影響。	1
(2) 它的成本（資金、人力、時間）是最低的。	1
3. (a) [Pascal 版本]	2
function call_random(N : integer) : integer; begin call_random := myrand(N) + 1; end;	
[C 版本]	
int call_random(int N) { return (myrand(N)+1); }	
[Visual Basic 版本]	
Private Function call_random(ByVal N As Integer) call_random = myrand(N) + 1 End Function	
[Java 版本]	
static int call_random(int N) { return (myrand(N)+1); }	
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ⓐ 調用及返回 ⓑ 返回值 </div>	
(b) 容易修改程式內的數值。／它令程式更易讀。／ 很容易改編程式，以便在該公司的其他大廈內使用。	1×2

(c) [Pascal 版本]

```
function closest : integer;
var
  i, cPos, cLift : integer;
begin
  cPos := MAXFLOOR;
  for i := 1 to LIFTTOTAL do
    if LiftPos[i] < cPos then
      begin
        cPos := LiftPos[i];
        cLift := i;
      end;
    closest := cLift;
  end;
end;
```

- ① i, cPos, cLift
- ① 初始化 cPos
- ① for 循環
- ① if 條件
- ① 找出最近的升降機
- ① 返回值
- ① 使用 MAXFLOOR 和 LIFTTOTAL

[C 版本]

```
int closest(void) {
  int i, cPos, cLift;
  cPos = MAXFLOOR;
  for (i=1; i<=LIFTTOTAL; i++)
    if (LiftPos[i] < cPos) {
      cPos = LiftPos[i];
      cLift = i;
    }
  return cLift;
}
```

[Visual Basic 版本]

```
Private Function closest() As Integer
  Dim i, cPos, cLift As Integer

  cPos = MAXFLOOR
  For i = 1 To LIFTTOTAL
    If LiftPos(i) < cPos Then
      cPos = LiftPos(i)
      cLift = i
    End If
  Next i
  closest = cLift
End Function
```

[Java 版本]

```
static int closest() {
  int i, cPos, cLift=0;
  cPos = MAXFLOOR;
  for (i=1; i<=LIFTTOTAL; i++)
    if (LiftPos[i] < cPos) {
      cPos = LiftPos[i];
      cLift = i;
    }
  return cLift;
}
```

分

- (d) (i) 與用戶面談（管理層，乘客）
問卷調查
觀察（例如實施電腦化前，觀察及取得升降機運作的第一手經驗）
審閱統計數據（例如等待升降機時間） 1x2

- (ii) 縮短乘客呼召升降機到地下的等待時間。 1

4. (a) **[Pascal 版本]** 1,4

```
(i) var A : arraytype;
(ii) while not eof(infile) do
    begin
        readln(infile, A[i])
        i := i + 1;
    end;
```

- ⓐ 考慮檔尾
- ⓑ 布爾算式
- ⓒ 讀取一行
- ⓓ 循環

[C 版本]

```
(i) char A[100][7]
(ii) while (fscanf(infile, "%s", A[i]) != EOF)
    i++;
```

可接受 6

[Visual Basic 版本]

```
(i) ByRef A(100) As String
(ii) While item <> Nothing
    A(i) = item
    i = i + 1
    item = infile.ReadLine()
End While
```

[Java 版本]

```
(i) String[] A
(ii) while (item != null) {
    A[i] = item;
    i+=1;
    item = infile.readLine();
}
```

(b) (i)

	A[0]	A[1]	A[2]	A[3]	A[4]
第二遍	chess	music	sports	art	ict
第三遍	art	chess	music	sports	ict

- (ii) 額外的第一遍 (j = 0) 是多餘的。 1
它只把第一個元素 (A[0]) 賦值至 Temp，然後把 Temp 賦值回 A[0]。
實際是跳過了步驟 4、5 和 6。 1

- (iii) 所有數據都是反序的 / 倒序排列。 1

- (c) (i) (1) false. 1
(2) B = flute, oboe. 1
(3) C = flute, oboe, basketball. 1

- (ii) ?- belongsto(art, Y). 1

- (iii) 邏輯語言專注於設立目標（「解決甚麼」），並使用關係解決問題（事實與規則關聯起來），而不是明確指出如何解決問題。 1