

下列程序段中，所有未有列出宣告 (declaration) 的變量，均假設已經適當地宣告。題目中的「整數」是指 32 位元有符號的變數 (Pascal: longint, C: int)。假設所有程序都正確地編譯，且沒有使用任何編譯器選項 (除 C 程序的"-o"選擇外)。

	格式	題目數	佔分
甲部	多項選擇題	25	25
乙部	填充題	6 (A-O)	20
總分			45

### 甲部 (25 分)

請為下列每題各選一個最適合的答案，然後把答案的字母 (A、B、C 或 D) 寫到答題紙對應的空格中。  
答對得一分，答錯不扣分。

- 以下哪段陳述是正確的?
  - 'A' 的美國信息交換標準碼 (ASCII 碼) 比 'a' 的為小。
  - ' ' (空白) 的美國信息交換標準碼 (ASCII 碼) 比 '0' 的為小。
  - 兩段陳述皆非
  - 只有 i
  - 只有 ii
  - i 和 ii
- 假設我們有一個存有無限數目襪子的盒子，我們每次可以拿出一隻隨機顏色的襪子，以下哪段陳述必然正確?
  - 若只有藍色和紅色襪子在盒子中，為保證至少取出一對有相同顏色的襪子，我們至少要取出 3 隻襪子。
  - 若有  $N$  種顏色的襪子在盒子中，為保證我們最少取出一對襪子相同顏色，我們至少要取出  $\frac{1}{2}N(N+1)$  隻襪子。
  - 若有  $N$  種顏色的襪子在盒子中，為保證我們最少取出有  $K$  隻襪子有相同顏色，我們至少要取出  $\frac{1}{2}N(N+1)(K-1)$  隻襪子。

取消

3. 假設以下的程序在某部電腦上運行需時兩秒。

**Pascal 版本**

```
sum := 0;
for i := 1 to 10000 do
begin
    sum := sum+i;
end;
```

**C 版本**

```
sum = 0;
for (i = 1; i <= 10000; i++) {
    sum = sum + i;
}
```

考慮以下程序段：

**Pascal 版本**

```
sum := 0;
for i := 1 to 20000 do
begin
    sum := sum+i;
    if (i > 0) then
    begin
        sum := sum-i;
        i := i+1;
    end;
end;
```

**C 版本**

```
sum = 0;
for (i = 1; i <= 20000; i++) {
    sum = sum + i;
    if (i > 0) {
        sum = sum - i;
        i++;
    }
}
```

以下哪項最能準確估計以上程序在同一電腦的運行時間？

- A. 2 秒
- B. 4 秒
- C. 6 秒
- D. 8 秒

4. 取消。原本的題見有遺漏。

在一個村落中，存在  $N$  ( $N \geq 4$ ) 間屋，它們之間互不相連，現在村中的長老決定在屋子間興建多條道路，以令屋子間的交通更方便。不過，由於一個神秘的習俗，三間屋子不能被道路直接連接。(即是若有 A、B、C 屋，且 A 和 B，B 和 C，C 和 A 都有路連接，壞事將會發生。) **道路不可以交叉。**

現在問最多能興建多少條路且不違反以上習俗？

- A.  $N$
- B.  $2N - 4$
- C.  $\frac{1}{2}(N - 1)(N - 2)$
- D. 道路的數目在不同  $N$  之下不同且沒有固定的公式

5. 在一個神秘的語言中，有些字會被認為是「神奇字」，「神奇字」中第一和最尾的字母必須是 'A' 且其他為 'B'，再者，有  $x$  個 'B' 的「神奇字」被稱為  $x$ -神奇字。

例如，"ABA" 是 1-神奇字，"ABBA" 是 2-神奇字，"ABBBA" 是 3-神奇字。

現在我們引入「神奇字串」，1-神奇字串是指 1-神奇字，"ABA"。2-神奇字串是在兩個 1-神奇字串中間插入一個 2-神奇字組成，是為"ABAABBAABA"。3-神奇字串是在兩個 2-神奇字串中間插入一個 3-神奇字組成，如此類推。如這過程無限持續，我們能生成一個 無限-神奇字串，其形態為：

ABAABBAABAABBBAABAABBAABAABBBBA...

其第一和第二字母是"A" 和"B"，以下哪個陳述為真？

- i. 第 50 個字母為"B"。
  - ii. 第 55 個字母為"B"。
  - iii. 第 60 個字母為"A"。
  - iv. 第 70 個字母為"A"。
- A. i 和 ii
  - B. i 和 iii
  - C. ii 和 iv
  - D. iii 和 iv
6. 很久以前，村民使用四種物品（米，椰菜，豬肉和水）來進行交易。
- 36 公斤的米能換取 17 公斤的椰菜，反之亦然。
  - 13 公斤的豬肉能換取 18 公斤的椰菜，反之亦然。
  - 12 加侖的水能換取 49 公斤的豬肉，反之亦然。

(你可以兌換非整數的貨物，如使用 18 公斤的米來換取 8.5 公斤的椰菜。

以下哪項貨物具有最高的價值？

- A. 101 公斤的米
- B. 48 公斤的椰菜
- C. 74 公斤的豬肉
- D. 18 加侖的水

7. 以下程序的輸出是?

**Pascal 版本**

```
var c:longint;
function f(a,b:longint):longint;
var tmp:longint;
begin
  tmp := 0;
  if (a = 1) then
    tmp := 1
  else if (b = 1) then
    tmp := 1
  else
    tmp := f(a-1, b) + f(a, b-1);
  f := tmp;
end;
begin
  c := f(2, 3) + f(3, 3);
  write(c);
end.
```

**C 版本**

```
#include <stdio.h>
int c;
int f(int a, int b){
  int tmp = 0;
  if (a == 1) {
    tmp = 1;
  } else if (b == 1) {
    tmp = 1;
  } else {
    tmp = f(a-1, b) + f(a, b-1);
  }
  return tmp;
}
int main() {
  c = f(2, 3) + f(3, 3);
  printf("%d", c);
  return 0;
}
```

- A. 5
- B. 7
- C. 9
- D. 11

8. 以下哪項有關數據結構的陳述句有可能是錯誤的?

- A. 二分搜索法可以對陣列使用。
- B. 冒泡排序可以對鏈表使用。
- C. 合併排序可以對堆疊使用。
- D. 線性搜索法能對隊列使用。

9. 有多少不同的方法在 4 到 20 間選擇三個不同的質數? (同一組合的不同次序視為相同)

- A. 20
- B. 70
- C. 120
- D. 170

10. 以下是運算符  $\diamond$  的真值表:

A	B	$A \diamond B$
真	真	真
真	假	真
假	真	真
假	假	假

簡化下列的布爾表達式，當中  $U$  和  $V$  都是布爾變量。

$$(\text{NOT } (U) \diamond (U \diamond V)) \text{ AND } V$$

- A. 真
- B. 假
- C.  $U$
- D.  $V$

11. 在一個派對內有很多參加者，他們之中有一對對的朋友。朋友的關係很特別，若  $A$  和  $B$  是朋友，且  $B$  和  $C$  是朋友，則  $A$  和  $C$  也是朋友。你知道以下資訊:

泰達米爾 和 塔里克 是朋友  
 易大師 和 悟空 是朋友  
 趙信 和 嘉文四世 是朋友  
 李星 和 蓋倫 是朋友  
 塔里克 和 趙信 是朋友

那麼在派對中最少有多少對朋友呢?

- A. 5
- B. 6
- C. 7
- D. 8

12. 以下排序算法的名稱是?

#### Pascal 版本

```
var
  a:array[0..4] of longint;
  i, j, k:longint;
begin
  read(a[0], a[1], a[2], a[3], a[4]);
  for i := 0 to 4 do
    begin
      j := a[i];
      k := i-1;
      while (k >= 0) do
        begin
          if (a[k] < j) then break;
          a[k+1] := a[k];
          k := k-1;
        end;
      a[k+1] := j;
    end
  end.
```

#### C 版本

```
#include <stdio.h>
int a[5];
int i, j, k;
int main() {
  scanf("%d%d%d%d%d", &a[0], &a[1], &a[2],
    &a[3], &a[4]);
  for (i = 0; i < 5; i++) {
    j = a[i];
    k = i-1;
    while (k >= 0) {
      if (a[k] < j) break;
      a[k+1] = a[k];
      k = k-1;
    }
    a[k+1] = j;
  }
  return 0;
}
```

- A. 普通排序
- B. 冒泡排序
- C. 插入排序
- D. 選擇排序

13. 以下程序段的輸出是?

#### Pascal 版本

```
var x:real;
...
x := 0.0000;
while (x <> 42.0000) do
begin
  x := x+0.1000
end;
write(x:0:4);
```

#### C 版本

```
double x = 0.0000;

while (x != 42.0000) {
  x = x+0.1000;
}
printf("%.4f", x);
```

- A. 43.0000
- B. 42.0000
- C. 41.9999
- D. 以上皆非

14. 考慮以下的程序，變數  $y$  是一個整數，且  $x$  被宣告為某種數據類別，其為  $[-200, 200]$  中的一個整數值。

**Pascal 版本**

```
y := -1;
if (x < 0) then
  y := 0
else if (x < 100) then
  y := 1
else if (x >= 100) then
  y := 2;
write(y);
```

**C 版本**

```
y = -1;
if (x < 0) {
  y = 0;
} else if (x < 100) {
  y = 1;
} else if (x >= 100) {
  y = 2;
}
printf("%d", y);
```

以下哪句陳述必然正確？

- A. 如  $x$  被宣告為 `double` (**Pascal: real**)，程序可能輸出 `-1`。
- B. 如  $x$  被宣告為 `int` (**Pascal: longint**)，程序可能輸出 `-1`。
- C. 如  $x$  的值為負數，程序可能會輸出 `0` 以外的數。
- D. 無論  $x$  被宣告成 `int` (**Pascal: longint**) 或 `double` (**Pascal: real**)，對於同一數值  $x$ ，輸出總是一樣。

15. 以下程序段將輸出什麼數值？

**Pascal 版本**

```
write(12 and 4, ' ', 10 or 21);
```

**C 版本**

```
printf("%d %d", 12 & 4, 10 | 21);
```

- A. 4 0
- B. 4 31
- C. 12 1
- D. 12 10

16. 以下哪種數據類型能夠作為 `switch (case-of)` 語句的控制變量？

- i. `char`
- ii. `double` (**Pascal: real**)

- A. 兩種類型皆非
- B. 只有 i
- C. 只有 ii
- D. i 和 ii

17. 大衛是一個非常聰明且十分喜歡曲奇餅的學生，每人都想把他們的曲奇餅送給大衛，企圖換取他的幫助以致在考試中合格，因此大衛說「如你有在班中最多曲奇餅，我會為你提供私人補習」。

假設愛麗絲和比伯分別在 A、B 班中擁有最多曲奇餅，他倆的曲奇餅總數為  $S$ ，以下哪段陳述必然正確？

- i. 愛麗絲和比伯會在考試中合格。
- ii.  $S$  大於愛麗絲和比伯的同學曲奇餅的總數。
- iii. 任意在 A、B 班中各自選擇一位同學，他們擁有的曲奇餅總數會比  $S$  少。
- iv. 對於 A 班中任意一位同學，其曲奇餅數目會少於  $S$  和比伯曲奇數目之間的差異。

- A. 四段陳述皆非
- B. 只有 i
- C. 只有 ii
- D. 只有 iii 和 iv

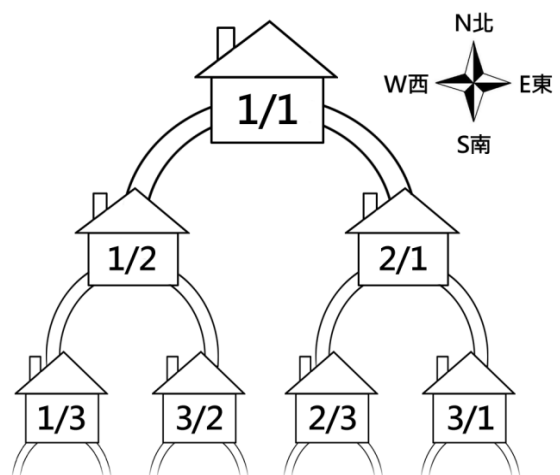
18. 現在有無窮多個五毫，一元和二元硬幣，你可以逐一把它們拾起直至剛好有 4.5 元，現在問有多少種不同的方法？(不同次序視為不同)

取消

19. 現在有一個無窮大的大村落，在每間屋子中，其名字為一個有理數  $(p/q)$ ，指向南方的兩間屋子，偏向西面的屋名為  $p/(p+q)$  且偏向東面的屋名為  $(p+q)/q$ ，屋子的名稱(有理數)永不會被約簡。

由於最北的屋子名稱為  $1/1$ ，村子的地圖和右圖相似：

由  $1/1$  開始只往南方行走，以下哪間屋子可在經過其他五間屋子之後到達？舉個例說，我們可經過一間屋子  $(1/2)$  後到達  $3/2$ 。



- A.  $17/9$
- B.  $17/10$
- C.  $13/15$
- D.  $11/8$



20. 以下程序的輸出是?

**Pascal 版本**

```
var
  i, j:longint;
  a:array[0..8] of longint =
    (2, 0, 3, 7, 5, 8, 6, 4, 2);
begin
  j := 1;
  for i := 0 to 6 do
    j := a[j];
  write(a[j]);
end.
```

**C 版本**

```
#include <stdio.h>
int i, j;
int a[9] = {2, 0, 3, 7, 5, 8, 6, 4, 2};
int main() {
  j = 1;
  for (i = 0; i <= 6; i++) {
    j = a[j];
  }
  printf("%d", a[j]);
  return 0;
}
```

- A. 2
- B. 0
- C. 8
- D. 5

21. 以下程序的輸出是?

**Pascal 版本**

```
var
  i, j:longint;
  a:array [0..8,0..1] of longint =
    ((1, 4), (3, 7), (6, 2), (7, 3), (0, 1),
     (8, 5), (8, 7), (4, 8), (4, 7));
begin
  j := 0;
  for i := 0 to 10007 do
    j := a[j][j mod 2];
  write(j);
end.
```

**C 版本**

```
#include <stdio.h>
int i, j;
int a[9][2] = {{1, 4}, {3, 7}, {6, 2}, {7, 3},
               {0, 1}, {8, 5}, {8, 7}, {4, 8}, {4, 7}};
int main() {
  j = 0;
  for (i = 0; i <= 10007; i++) {
    j = a[j][j%2];
  }
  printf("%d", j);
  return 0;
}
```

- A. 2
- B. 8
- C. 0
- D. 3

22. 以下程序的輸出是?

**Pascal 版本**

```
function fact(n:longint):longint;
begin
  if (n = 0) then
    fact:=2
  else if ((n mod 3) = 0) then
    fact:=fact(n-1)*3
  else if ((n mod 2) = 0) then
    fact:=fact(n-1)*2
  else
    fact:=fact(n-1)*1
end;
begin
  write(fact(10))
end.
```

**C 版本**

```
#include <stdio.h>
int fact(int n) {
  if (n == 0) {
    return 2;
  } else if (n%3 == 0) {
    return fact(n-1)*3;
  } else if (n%2 == 0) {
    return fact(n-1)*2;
  } else {
    return fact(n-1)*1;
  }
}
int main() {
  printf("%d", fact(10));
  return 0;
}
```

- A. 864
- B. 1728
- C. 2592
- D. 以上皆非

23. 編譯器能將源代碼編譯成程式。當然，一個編譯器是需要用來編譯編譯器的。(編譯器是一個程式)

設編譯器 A 的源代碼是代碼 A。當我們使用編譯器 A 編譯代碼 A 時，我們將得到編譯器 A。設編譯器 B 的源代碼是代碼 B。當我們使用編譯器 A 編譯代碼 B 時，我們將得到編譯器 B。以下那些陳述必定正確？

注意：如果兩個檔案的所有位元相同，那麼兩個檔案是相等的。

- i. 編譯器 A 和編譯器 B 是相同編程語言的編譯器。
- ii. 如果我們使用編譯器 B 編譯代碼 A，其程式輸出將相等於編譯器 A。
- iii. 如果我們使用編譯器 B 編譯代碼 B，其程式輸出將相等於編譯器 B。

- A. 三段陳述皆非
- B. 只有 i
- C. 只有 ii
- D. 只有 iii

24. 設代碼 A 現在被更改到一個新的版本 -- 代碼 C。當代碼 C 被編譯器 A 編譯，我們將得到編譯器 C。編譯器 A 和編譯器 C 是相同編程語言的編譯器。編譯器 C 總是能比編譯器 A 編譯出更快的程式 (即當相同的代碼分別被編譯器 A 和編譯器 C 編譯(及無編譯錯誤)，然後將相同的輸入分別給予兩個編譯器所輸出的程式，程式的輸出是相同的(設無運行錯誤)，而編譯器 C 給出的程式總是比編譯器 A 給出的程式有更短的運行時間(設兩個程式均會終結)。這個特性對於所有的輸入均是相同的)。當我們使用編譯器 C 編譯代碼 A，我們將得到編譯器 A2。當我們使用編譯器 C 編譯代碼 C，我們將得到編譯器 C2。以下哪段陳述必定正確?

- i. 編譯器 A2 比編譯器 A 快。(即編譯器 A2，相對於編譯器 A 來言，只需較少時間編譯任何代碼)
- ii. 編譯器 A2 比編譯器 C 快。(即編譯器 A2，相對於編譯器 C 來言，只需較少時間編譯任何代碼)
- iii. 編譯器 C2 比編譯器 A 快。(即編譯器 C2，相對於編譯器 A 來言，只需較少時間編譯任何代碼)

- A. 只有 i
- B. 只有 iii
- C. 只有 i 和 iii
- D. i, ii 和 iii

25. 考慮以下程序段:

**Pascal 版本**

```
[I]
begin
  write('1');
  [J]
  begin
    [K]
    begin
      write('3')
    end;
    write('2')
  end
end
```

**C 版本**

```
[I] {
  printf("1");
  [J] {
    [K] {
      printf("3");
    }
    printf("2");
  }
}
```

我們把 [I], [J] 和 [K] 各自代入以下一行:

**Pascal 版本**

```
L: for i := 0 to 1 do
M: for j := 0 to 2 do
N: for k := 0 to 3 do
```

**C 版本**

```
L: for (i = 0; i <= 1; i++)
M: for (j = 0; j <= 2; j++)
N: for (k = 0; k <= 3; k++)
```

哪個組合能生成最小的輸出? (視輸出為十進制數字)

- |    |     |     |     |
|----|-----|-----|-----|
|    | [I] | [J] | [K] |
| A. | L   | M   | N   |
| B. | M   | N   | L   |
| C. | N   | L   | M   |
| D. | N   | M   | L   |

**乙部 (20 分)**

下列各空格分別命名為 A 至 O，請在答題紙上對應的地方填上答案。  
除非另外註明，否則答對得兩分，答錯不扣分。

**注意:**

- (1) 答案不可以包括 C 語言的 ? : 運算元。
- (2) 除非適當的函數庫已被引用，否則答案不可以包括任何函數庫內的函數。
- (3) 答題紙上每個小格只可填上一個字符。
- (4) 答案長度不得多於該題提供的小格數目。

1. 將以下的十六進制數字轉換成八進制。

FACADE(16) →     A    (8) (1 分)

將以下的二進制數字轉換成十六進制。

1010010000111101(2) →     B    (16) (1 分)

2. 「九達通卡」是一種電子貨幣。把九達通卡的餘額輸入以下程式，它就能自動檢查餘額是否正數。若非如此，程式會(重覆)為卡增值\$200 直至餘額為正數。然後它會輸出最終餘額。你可假設輸入的數字 x 必為 [-2000, 2000] 的整數。完成這程序。

**Pascal 版本**

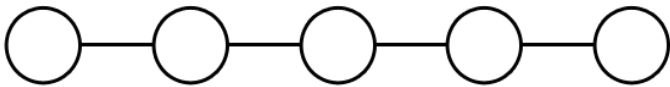
```
var
  x:longint;
begin
  read(x);
  if (x > 0) then
    write(x)
  else
    write(    C    );
end.
```

**C 版本**

```
#include <stdio.h>
int x;
int main(){
  scanf("%d", &x);
  if (x > 0)
    printf("%d", x);
  else
    printf("%d",     C    );
  return 0;
}
```

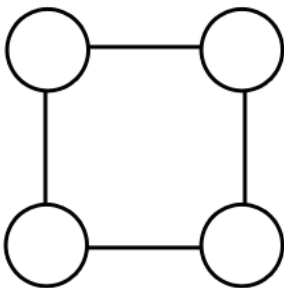
C: 2 分

3. 對於一個圖，我們把 1, 2, 3 或 4 放在每個圓內，以令每個相鄰的圓內，有不同的數字。我們叫這種組態為"4-數字分布"，兩個這樣的組態如在任何一對相應數字不相同，便稱為不同的組態。



問題: 對於以上的圖，總共有多少可能的 4-數字分布組態?

答案:   D   (1 分)



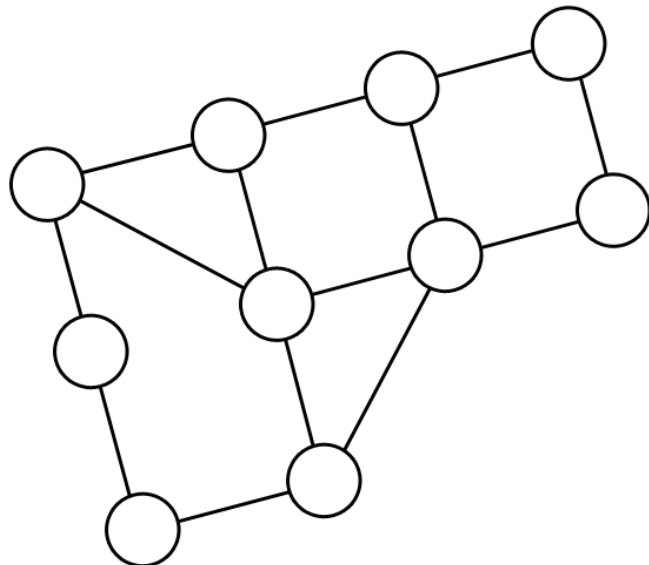
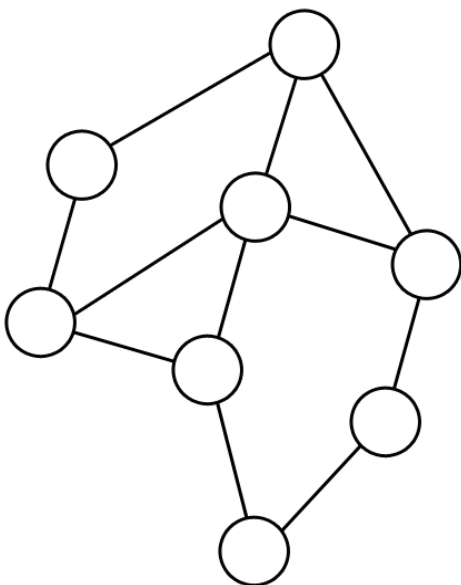
對於以上的圖，總共有多少可能的 4-數字分布組態?

答案:   E   (1.5 分)

已知左圖有 2,332 種不同的 4-數字分布組態。

對於右方的圖，總共有多少可能的 4-數字分布組態?

答案:   F   (2 分) 取消。原本的題目的數字不正確。



4. 考慮以下程序:

#### Pascal 版本

```

11 var
12   i, n:longint;
13 begin
14   write('Input integer n: ');
15   read(n);
16   for i := 1 to 45 do
17   begin
18     if (i*(i+1) = n*2) then
19     begin
20       write('n is a triangular number');
21     end
22   end;
23   write('n is not a triangular number');
24 end.
```

#### C 版本

```

51 #include <stdio.h>
52 int n, i;
53 int main() {
54   printf("Input integer n: ");
55   scanf("%d", &n);
56   for (i = 1; i <= 45; i++) {
57     if (i*(i+1) == n*2) {
58       printf("n is a triangular number");
59     }
60   }
61   printf("n is not a triangular number");
62   return 0;
63 }
```

以上程序判斷輸入  $n$  是否一個三角形數。你可以假設  $n$  為  $[1,1000]$  內的整數。可是，這程序有一錯誤，但只要更改其中一行便可修正。找出該錯誤的類型（編譯、運行或邏輯錯誤）和行數，然後重寫。

錯誤類別:   G   (1 分, 剔選  合適答案)

行數:   H   (1 分)

重寫:   I   (2 分)

5. 假設函數/程序 `swap(i, j)` 交換 `a[i]`和 `a[j]`的值，例如若 `a[3]=5`、`a[6]=1`，在 `swap(3, 6)` 後，`a[3]` 會變成 1、`a[6]` 變成 5。完成以下程序段，以令其能升序排序陣列 `a[9]`。 (**Pascal**: `a[0..8]`)。

**Pascal 版本**

```
p := 0;
while (p < 8) do
begin
  if a[p] > a[p+1] then
  begin
        J    ;
    if (    K1    ) then
          K2    ;
  end
  else
  begin
        L    ;
  end
end;
end;
```

**C 版本**

```
p = 0;
while (p < 8) {
  if (a[p] > a[p+1]) {
        J    ;
    if (    K1    )
          K2    ;
  } else {
        L    ;
  }
}
```

J: 1 分      K1 和 K2: 2 分若全對      L: 1 分

6. 以下程序的輸出是?

**Pascal 版本**

```
var
  s:string;
  i, j:longint;
begin
  s := '..>..<.>..';
  for i := 1 to 3 do
  begin
    for j := 1 to 9 do
    begin
      if (s[j+1] = '<') then
      begin
        s[j] := '<';
        s[j+1] := '.';
      end;
      if (s[10-j] = '>') then
      begin
        s[11-j] := '>';
        s[10-j] := '.';
      end;
    end;
  end;
  writeln(s);
end.
```

**C 版本**

```
#include <stdio.h>
int main() {
  int i, j;
  char s[] = "..>..<.>..";
  for (i = 1; i <= 3; i++) {
    for (j = 0; j <= 8; j++) {
      if (s[j+1] == '<') {
        s[j] = '<';
        s[j+1] = '.';
      }
      if (s[8-j] == '>') {
        s[9-j] = '>';
        s[8-j] = '.';
      }
    }
    printf("%s\n", s);
  }
  return 0;
}
```

第 1 行:     M     (1 分)  
 第 2 行:     N     (1 分)  
 第 3 行:     O     (1.5 分)