

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)。

|       | Format | # Questions | Total 分 |
|-------|--------|-------------|---------|
| 甲部(一) | 多項選擇題  | 25          | 25      |
| 甲部(二) | 填充題    | 7 (A-J)     | 20      |
| 總分    |        |             | 45      |

### 甲部 (20 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。

1. 給予以下五條等式:

$$a + d = 10$$

$$c + e = -3$$

$$b + e = 1$$

$$d + c = 5$$

$$b + f = 2$$

下列哪個表達式的數值可以被計算出來？

i.  $a + f$

ii.  $b + d$

iii.  $d + e$

- A. 只有 iii  
 B. 只有 i 和 ii  
 C. 只有 ii 和 iii  
 D. 全部皆可以

2. 以下程序的輸出是？

#### Pascal 版本

```
var
  sum1, sum2: double;
  i: longint;
begin
  sum1 := 0.0;
  sum2 := 0.0;
  for i := 1 to 6 do
  begin
    sum1 := sum1 + 1.0 / i;
    sum2 := sum2 - 1.0 / i;
  end;
  if (sum1 = 2.45) then write('A');
  if (sum1 + sum2 = 0.0) then
  write('B');
end.
```

#### C 版本

```
#include <stdio.h>
double sum1, sum2;
int i;
int main() {
  sum1 = 0.0;
  sum2 = 0.0;
  for (i = 1; i <= 6; i++) {
    sum1 = sum1 + 1.0f / i;
    sum2 = sum2 - 1.0f / i;
  }
  if (sum1 == 2.45) printf("A");
  if (sum1 + sum2 == 0.0) printf("B");
  return 0;
}
```

- A. A
- B. B
- C. AB
- D. 無任何輸出

3. 689 是個極為美麗的數字，因為它即使旋轉了 180 度仍然是 689。101 又是一個長度為 3 的美麗數字，很可惜 2015 不是一個長度為 4 的美麗數字。留意美麗數字並不合前導零。總共有多少個長度為 2015 的美麗數字？答案的個位數是？

- A. 2
- B. 0
- C. 1
- D. 5

對於 4 至 5 題，考慮以下程序：

**Pascal 版本**

```
var
  max, i: longint;
  a: array[0..4] of longint;
begin
  max := 0;
  for i := 0 to 4 do
    read(a[i]);
  for i := 0 to 3 do
    if (a[i] + a[i+1] > max) then
      max := a[i] + a[i+1];
  write(max);
end.
```

**C 版本**

```
#include <stdio.h>
int max, n, i, a[5];
int main() {
  max = 0;
  for (i = 0; i < 5; i++)
    scanf("%d", &a[i]);
  for (i = 0; i < 4; i++)
    if (a[i] + a[i+1] > max)
      max = a[i] + a[i+1];
  printf("%d", max);
  return 0;
}
```

4. 如果輸入 5 -6 2 -5 3，這程式會輸出甚麼？

- A. -2
- B. -1
- C. 0
- D. 8

5. 以下哪一項輸入會令程式輸出最大的數字？

- A. -2100000000 2000000000 999999999 -1500000000 1234567890
- B. 5 1000 -300 0 5
- C. -500 50000 -2000000000 -1000000000 800
- D. -800000 50000 -999999999 100000 5000

6. 迴文是從前後兩個方向讀都相同的字串。例如，"abcba"是一個迴文，而"accda"不是。一段字串的子字串是這字串的任何連續線段。例如，"a"、"b"、"c"、"ab"、"bc"、"abc"都是"abc"的子字串，而"ac"不是。

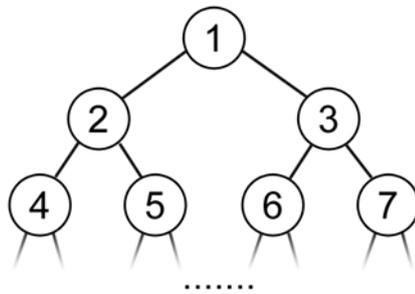
有多少長度剛好為 4 的字串並不存在任何長度大於 1 的迴文子字串？字串由"a"至"z"組成。例如，"abcd"是其中一個符合條件的字串，但"aabc"不是。("aa"為一迴文)

- A. 374400
- B. 390000
- C. 405600
- D. 422500

7. 愛麗絲剛剛上電腦課學會了壓縮檔案。她把 flower.jpg (2MB) 壓縮成 flower.zip (1MB)，然後再把 flower.zip 壓縮成 flower.zip.zip。一般情況下，以下哪項是該雙重壓縮檔案大小的最佳估算？

- A. 0MB
- B. 0.5MB
- C. 1MB
- D. 2MB

8. 以下將使用根值為 1 的二元樹。樹上每個值為  $N$  的節點有一個值為  $N \times 2$  的左兒子和一個值為  $N \times 2 + 1$  的右兒子。下圖是此樹開首數層:



現在定義一節點的祖先為它自己及它父節點的祖先。例如 7 的祖先包括 1 和 3。

兩節點  $v$  與  $w$  的共同祖先就是同時身為  $v$  的祖先和  $w$  的祖先的節點。例如 6 和 7 的共同祖先是 1 和 3。最低（最深層）的一個共同祖先稱為最低共同祖先。所以 6 和 7 的最低共同祖先是 3。

2338 和 2015 的最低共同祖先是？

- A. 1
- B. 2
- C. 1024
- D. 2048

9. 在一個 4 x 4 的棋盤上,每一格都有放上黑色或白色的棋子。棋盤上兩格若分享一條共同邊,則視為相鄰的兩格。

每一隻棋子均擁有單數數目同色且相鄰的棋子,下面是一個符合規則的例子。

```
oxxx
ooxo
oxoo
xxxx
```

現在問總共有多少種符合規則的方法填充以下填盤? ("x"、"o"、"." 分別代表黑色棋子、白色棋子和需要被填充的格子)

```
x.oo
....
....
....
```

- A. 1
- B. 2
- C. 8
- D. 16

10. 以下程序的輸出是?

**Pascal 版本**

```
var
  a: array[0..3] of longint;
  i, t: longint;
begin
  a[0] := 3; a[1] := 2; a[2] := 9; a[3] :=
5;
  t := a[0];
  for i := 1 to 3 do
  begin
    t := ((t * 2) xor (a[i]));
  end;
  write(t);
end.
```

**C 版本**

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

- A. 2
- B. 3
- C. 5
- D. 7

11. 以下程序的輸出是？

**Pascal 版本**

```
var
  ans, i, j: longint;
begin
  ans := 0;
  for i := 1 to 2003 do
  begin
    j := i;
    while j > 0 do
    begin
      ans := ans + j mod 10;
      j := j div 10;
    end;
  end;
  write(ans);
end.
```

**C 版本**

```
#include <stdio.h>
int ans, i, j;
int main() {
  ans = 0;
  for(i = 1; i <= 2003; i++){
    j = i;
    while(j > 0){
      ans = ans + j % 10;
      j = j / 10;
    }
  }
  printf("%d", ans);
  return 0;
}
```

- A. 13510
- B. 14510
- C. 27014
- D. 28014

對於 12 至 13 題，考慮以下情況:

愛麗絲和鮑伯正在玩一個二人遊戲 - X，在遊戲當中，玩家會輪流行動，直至某玩家作出了決勝的一步，他便會勝出並完結遊戲。現在，他們發現了無論後手如何對應，先手(第一個作出行動的玩家)都有辦法獲勝，他們因此感到沉悶。為此，他們新增了以下數道規則:

- 舊版遊戲將會重覆 N 回合 (每一回合和 X 完全相同)
- 第一回合的先手是愛麗絲
- 第 k 回合( $1 \leq k < N$ ) 的輸家將會成為第 k + 1 回合的先手
- 第 N 回合的贏家是整場遊戲的優勝者

12. 現在已知雙方皆不想落敗，在以下哪種情況中愛麗絲會勝出？

N =

- i. 5
  - ii. 10
  - iii. 13
- A. 只有 ii
  - B. 只有 i 和 iii
  - C. 無從判斷
  - D. 以上所有

13. 他們仔細研究過這遊戲後，發現在每一回合中，無論後手如何對應，先手都能夠令自己輸掉該回合。

現在已知雙方皆不想落敗，在以下哪種情況中愛麗絲會勝出？

N =

- i. 5
  - ii. 10
  - iii. 13
- A. 只有 ii
  - B. 只有 i 和 iii
  - C. 無從判斷
  - D. 以上所有

14. 以下程序的輸出是？

#### Pascal 版本

```
var
  i, x: longint;
begin
  x := 0;
  for i := 53 to 102 do
    x := x xor i;
  write(x)
end.
```

#### C 版本

```
#include <stdio.h>
int i, x;
int main() {
  x = 0;
  for(i = 53; i <= 102; i++)
    x = x ^ i;
  printf("%d", x);
  return 0;
}
```

- A. 83
- B. 0
- C. 1
- D. 52

15. 以下程序的目的是把正整數 n 分解成質數的乘積。

#### Pascal 版本

```
var
  n, k: longint;
begin
  read(n);
  k := 2;
  while (k <= n) do
  begin
    if (n mod k = 0) then
    begin
      n := n div k;
      if (n > 1) then
        write(k, ' * ');
      else
        write(k)
    end;
    k := k+1;
  end
end.
```

#### C 版本

```
#include <stdio.h>
int k, n;
int main(){
  scanf("%d", &n);
  k = 2;
  while (k <= n) {
    if (n%k == 0) {
      n = n/k;
      if (n > 1)
        printf("%d * ", k);
      else
        printf("%d", k);
    }
    k = k+1;
  }
}
```

但這程序有一邏輯錯誤，對於以下哪個(些)輸入  $n$ ，程序不能輸出預期的答案？

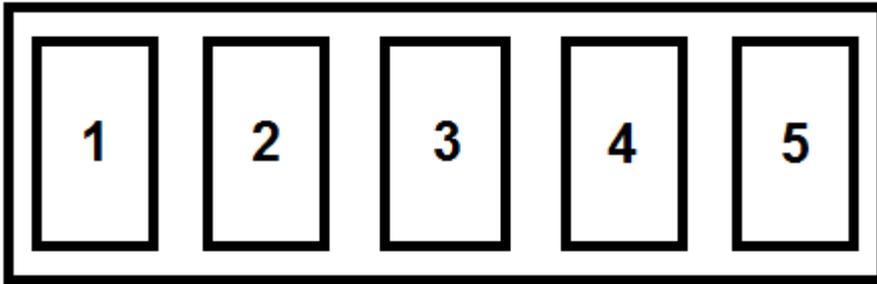
- i. 15
- ii. 16
- iii. 17
- iv. 18

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

16. 在辦公室中有一機器 - X，其用途是安全地儲存  $N$  份文件，當中每一份文件都會獨立地儲存在機器的某一格中。

現在我們稱機器最左的一格為第一格，最右的一格為第  $N$  格。

以下是當機器 X 儲存著五份文件時的內部結構，在這圖中，第  $p$  份文件正被儲存在第  $p$  格。當一個職員向 X 查詢第  $k$  格中的文件(不一定是第  $k$  份文件)，X 會先把文件移至最接近該格的一端，並在職員完成查詢後把文件歸還至原位，過程共消耗  $\min(N - k + 1, k)$  單位能量。(  $\min(a, b)$  會回傳  $a, b$  中的最小值)



以上圖作例子：當  $k = 2$ ，X 會使用 2 單位能量把文件移至左端並歸還。

當  $k = 5$ ，X 會使用 1 單位能量把文件移至右端並歸還。

X 的發明家剛記錄了每天中各份文件的使用頻率

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 文件 | 1 | 2 | 3 | 4 | 5 |
| 頻率 | 4 | 2 | 5 | 1 | 6 |

假設以上頻率每天皆相同，如果他能夠重新編排文件的次序，每天最少的單位能量消耗是？

- A. 26
- B. 27
- C. 28
- D. 29

17. 一班中有二十人，當中每人都坐在  $4 \times 5$  網格中的不同位置，現在老師想把全班分成三部分來進行團體習作。由於老師坐在眾同學前，他有四種方法進行分組:由前方最左面／最右面的同學開始以兩個方向進行不停點數 1, 2, 3, 1, 2, 3... (如圖所示)，直至每人都被分配到一個組號。

明顯地，第三組將會分配到最少的工作人數。現在問有多少個座位無論在何種分配方法當中都會被分配到第三組？

```

12312  12...  ...21  21321
3..... 23...  ...32  ...13
.....  3.....  ....3  .....
.....  1.....  ....1  .....
    
```

- A. 0
- B. 2
- C. 4
- D. 6

18. 以下程序的輸出是？

**Pascal 版本**

```

var
  tmp, cnt, i: longint;
begin
  tmp := 0;
  cnt := 0;
  i := 1;
  while (i < 100) do
  begin
    tmp := tmp * i;
    tmp := tmp + i;
    tmp := tmp mod 1000;
    if (tmp mod 5 = 0) then
      cnt := cnt + 1;
    i := i + 1;
  end;
  write(cnt);
end.
    
```

**C 版本**

```

#include <stdio.h>
int tmp, cnt, i;
int main() {
  tmp = 0;
  cnt = 0;
  i = 1;
  while (i < 100) {
    tmp = tmp * i;
    tmp = tmp + i;
    tmp = tmp % 1000;
    if (tmp % 5 == 0)
      cnt = cnt + 1;
    i = i + 1;
  }
  printf("%d", cnt);
  return 0;
}
    
```

- A. 20
- B. 29
- C. 39
- D. 40

19. 以下程序的輸出是？

**Pascal 版本**

```
var
  k, x, y, tx, ty: longint;
  a: array[0..2, 0..4] of longint =
    ((11, 3, 7, 15, 1), (16, 62, 53, 44, 37),
    (10, 12, 11, 31, 22)) ;
begin
  k := 127;
  x := 1;
  tx := 1;
  y := 0;
  ty := 0;
  while k>0 do
  begin
    dec(k);
    x := (a[tx][ty]+tx) mod 3;
    y := (a[tx][ty]+ty) mod 5;
    tx := x;
    ty := y;
  end;
  write(x, ' ', y);
end.
```

**C 版本**

```
#include <stdio.h>
int k, x, y, tx, ty, a[3][5] = {{11, 3,
7, 15, 1}, {16, 62, 53, 44, 37}, {10,
12, 11, 31, 22}};
int main(){
  k = 127;
  x = 1;
  tx = 1;
  y = 0;
  ty = 0;
  while(k>0){
    k--;
    x = (a[tx][ty] + tx) % 3;
    y = (a[tx][ty] + ty) % 5;
    tx = x;
    ty = y;
  }
  printf("%d %d", x, y);
  return 0;
}
```

- A. 04
- B. 03
- C. 13
- D. 23

20. 假設我們不斷拋硬幣，直至連續兩次出現"公"便停止，平均需要拋硬幣多少次我們才會停止？

- A. 4.00
- B. 4.50
- C. 5.25
- D. 6.00

21. 變數  $a$  和  $b$  是整數，以下哪項表達式與  $a > b$  等價？

- i.  $a - b > 0$
- ii.  $a + 1 > b + 1$
- iii.  $a * 1 > b * 1$
- iv.  $a * -1 < b * -1$

- A. 只有 iii
- B. 只有 i 和 ii
- C. 只有 iii 和 iv
- D. 以上皆是

對於 22 至 23 題，考慮以下情況：

在英雄聯盟遊戲中，一支隊伍中有五名成員。現在李星、飛斯、汎、瑟雷西、伊澤瑞爾組成了一支隊伍。

隊伍中有四個崗位，分別是：打野，上路，中路，下路。  
每名成員必須擔當剛好一個崗位。

22. 這支隊伍需要一名打野、一名上路、一名中路、兩名下路。  
請問有多少種分配崗位的方法？

- A. 60
- B. 120
- C. 240
- D. 600

23. 由於出現了各種新戰術，現在每個崗位可由最少一人、最多二人擔任。  
請問有多少種分配崗位的方法？

- A. 60
- B. 120
- C. 240
- D. 600

24. 以下程序的輸出是？

#### Pascal 版本

```
var
  n, count: longint;
begin
  n := 75290635;
  count := 0;
  while (n > 0) do
  begin
    while (n mod 10 <> 0) do
    begin
      n := n - 1;
      if (n mod 10 = 4) then
        continue;
      count := count + n mod 10;
    end;
    n := n div 10;
  end;
  write(count);
end.
```

#### C 版本

```
#include <stdio.h>
int n, count;
int main() {
  n = 75290635;
  count = 0;
  while (n > 0) {
    while (n % 10 != 0) {
      n = n - 1;
      if (n % 10 == 4)
        continue;
      count = count + n % 10;
    }
    n = n / 10;
  }
  printf("%d", count);
  return 0;
}
```

- A. 32
- B. 37
- C. 76
- D. 113

25. 以下程序的輸出是？

**Pascal 版本**

```
var
  a:array[0..4] of longint;
  i, j, temp: longint;
begin
  a[0] := 1; a[1] := 2; a[2] := 4; a[3] :=
3; a[4] := 5;
  for i := 1 to 5102 do
  begin
    temp := a[0];
    for j := 0 to 3 do
      a[j] := a[j+1];
    a[4] := temp;
  end;
  for i := 0 to 4 do
    write(a[i], ' ');
  end.
```

**C 版本**

```
#include <stdio.h>
int a[5];
int i, j, temp;
int main() {
  a[0] = 1; a[1] = 2; a[2] = 4; a[3] =
3; a[4] = 5;
  for (i = 1; i <= 5102; i++) {
    temp = a[0];
    for (j = 0; j <= 3; j++)
      a[j] = a[j+1];
    a[4] = temp;
  }
  for (i = 0; i <= 4; i++)
    printf("%d ", a[i]);
  return 0;
}
```

- A. 1 2 4 3 5
- B. 2 1 5 4 3
- C. 4 3 5 1 2
- D. 5 1 2 4 3

甲部完結

## 乙部 (20 分)

下列各空格分別命名為 A 至 J，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. NAND(非-與)函數有很多用途。NAND 使用兩個輸入，而它的真值表如下

| A | B | A NAND B |
|---|---|----------|
| 真 | 真 | 假        |
| 真 | 假 | 真        |
| 假 | 真 | 真        |
| 假 | 假 | 真        |

不妨以 Q 表示 NAND 函數。所以，兩個布爾表達式 X, Y 的 NAND 函數為  $(XQY)$ ，此結果也是個布爾表達式。A 和 B 自己是一個布爾表達式。例如，A 的非(NOT)-函數可以用  $(AQA)$  表示。

注意以上例子中的括號是必要的，括號不齊全的答案將不予評分

給予兩個輸入 A 和 B，以 A、B、括號和 Q (NAND 函數) 表達以下真值表。

| A | B | 結果 |
|---|---|----|
| 真 | 真 | 真  |
| 真 | 假 | 假  |
| 假 | 真 | 假  |
| 假 | 假 | 假  |

答案: \_\_\_\_\_ A \_\_\_\_\_ (2 分)

給予兩個輸入 A 和 B，以 A、B、括號和 Q (NAND 函數) 表達以下真值表。

| A | B | 結果 |
|---|---|----|
| 真 | 真 | 真  |
| 真 | 假 | 假  |
| 假 | 真 | 假  |
| 假 | 假 | 真  |

答案: \_\_\_\_\_ B \_\_\_\_\_ (2 分)

2. 以下程式讀入一整數  $n$  然後輸出一個  $(2n+1) \times (2n+1)$  網格。

**Pascal 版本**

```
var
  n, i, j: longint;
begin
  read(n);
  for i := 0 to 2*n do
  begin
    for j := 0 to 2*n do
      if (_____ C, D _____) then
        write('*')
      else
        write(' ');
    writeln();
  end;
end.
```

**C 版本**

```
#include <stdio.h>
int main() {
  int n, i, j;
  scanf("%d", &n);
  for (i = 0; i <= 2*n; i++) {
    for (j = 0; j <= 2*n; j++) {
      if (_____ C, D _____)
        printf("*");
      else
        printf(" ");
    }
    printf("\n");
  }
  return 0;
}
```

如果輸入  $n$  是 4，完成程序使得輸出是

```
*      *
 *     *
  *    *
   *   *
    *  *
     *
    *  *
   *   *
  *    *
 *     *
*      *
```

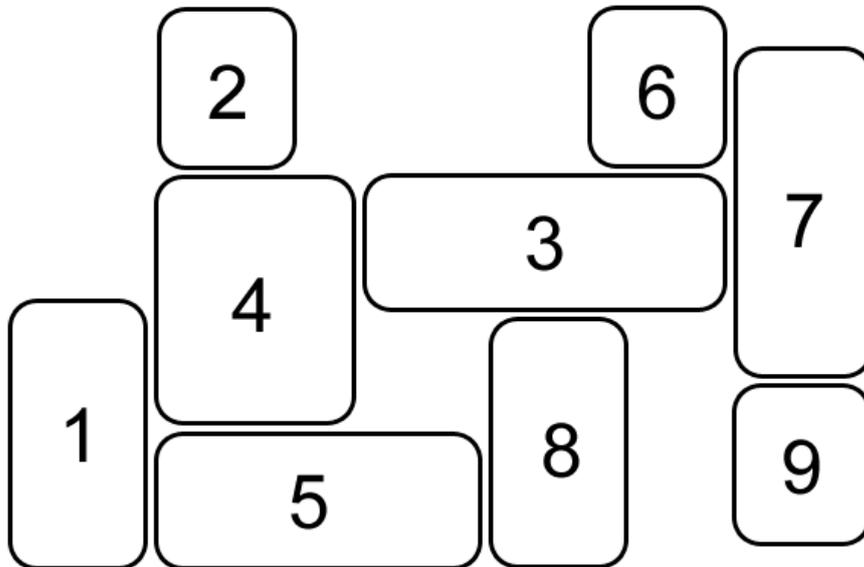
答案: \_\_\_\_\_ C \_\_\_\_\_ (2 分)

如果輸入  $n$  是 5，完成程序使得輸出是

```
*****
*      *
*     * *
*      *
*     * *
* * * * *
*     * *
*      *
*     * *
*      *
*****
```

答案: \_\_\_\_\_ D \_\_\_\_\_ (2 分)

3. 現在有一些冰塊浮在水面上。長方形代表冰塊，而長方形之間的窄縫代表你能從兩個相連的冰塊中走過。愛麗絲在其中一塊冰塊上。由於冰塊不斷溶化，她需要從一塊冰塊移動到另一塊冰塊上。在她離開某塊冰塊後，那塊冰塊將會溶化，而愛麗絲將不能再站在其上。最後她會停留在其中一塊冰塊上無路可走。愛麗絲的敵人鮑伯想知道所有愛麗絲可能最後停下的地方，請寫下那些冰塊的編號。



答案:       E       (2 分)

4. 在國際象棋中，「象」可以向四個斜線方向攻擊。現在知道在一塊 8 x 8 的棋盤上只有兩隻象，第一隻位於第 r1 行第 c1 列，第二隻位於第 r2 行第 c2 列。以下函數被傳入兩隻象的座標後，如果存在一個位置可以同時被兩隻象所攻擊，其回傳值為 0 (Pascal : false)，否則回傳值為 1 (Pascal : true)，現在請完成以下函數。

函數 check 符合：  
 $1 \leq r1, r2, c1, c2 \leq 8$

**Pascal 版本**

```
function check(r1, c1, r2, c2 :
longint) : boolean;
begin
  check :=       F      ;
end
```

**C 版本**

```
int check(int r1, int c1, int r2, int
c2) {
  return       F      ;
}
```

答案:       F       (2 分)

5. 考慮以下程序:

**Pascal 版本**

```

var
  a: array[0..9] of longint =
    (2 + 1, 0 + 5, 6 + 0, 0, 0, 1 + 9,
     8 + 2, 0, 9 + 6, 5 + 8);
procedure f(x, y: longint);
var
  i, tmp: longint;
begin
  for i := 1 to 7 do
  begin
    write(x);
    tmp := y;
    y := a[y] - x;
    x := tmp;
  end;
end;
procedure G;
begin
  _____ G _____
end;
procedure H;
begin
  _____ H _____
begin
end;

```

**C 版本**

```

#include <stdio.h>

int a[10] = {2 + 1, 0 + 5, 6 + 0, 0, 0,
1 + 9, 8 + 2, 0, 9 + 6, 5 + 8};
void f(int x, int y) {
  int i, tmp;
  for (i = 0; i < 7; i++) {
    printf("%d", x);
    tmp = y;
    y = a[y] - x;
    x = tmp;
  }
}

void G() {
  _____ G _____
}

void H() {
  _____ H _____
}

```

填充 \_\_\_\_\_ G \_\_\_\_\_ 使得呼叫 G() 將輸出 2015986。(2 分)

填充 \_\_\_\_\_ H \_\_\_\_\_ 使得呼叫 H() 將輸出 5102689。(2 分)

6. 考慮以下程序:

**Pascal 版本**

```

var
  i: longint;
begin
  for i := 0 to 15 do
  begin
    write(chr(_____ I _____));
    if (i mod 4 = 3) then
      writeln()
  end;
end.

```

**C 版本**

```

#include <stdio.h>
int main() {
  int i;
  for (i = 0; i <= 15; i++) {
    printf("%c", _____ I _____);
    if (i % 4 == 3)
      printf("\n");
  }
  return 0;
}

```

完成程序使得輸出為:

dhlp  
cgko  
bfjn  
aeim

答案: \_\_\_\_\_ I \_\_\_\_\_ (2 分)

請留意 Pascal 版本已包含 chr()

7. 為避免 Pascal 中的變數命名衝突，原本題目的變數  $d$  更改為變數  $x$ 。由於在原本題目的中文考卷中沒有定義  $d$ ，因此作答“編譯錯誤”的考生亦被給予分數。

以下程序的輸出是？

#### Pascal 版本

```
var
  n, x, A, B, C, D : longint;
begin
  n := 2015;
  x := 3;
  A := (n div x) * x + n mod x;
  B := (n div (-x)) * (-x) + n mod (-x);
  C := ((-n) div x) * x + (-n) mod x;
  D := ((-n) div (-x)) * (-x) + (-n) mod
(-x);
  write(A + B + C + D)
end.
```

#### C 版本

```
#include <stdio.h>
int n, x, A, B, C, D;
int main() {
  n = 2015;
  x = 3;
  A = (n / x) * x + n % x;
  B = (n / (-x)) * (-x) + n % (-x);
  C = ((-n) / x) * x + (-n) % x;
  D = ((-n) / (-x)) * (-x) + (-n) % (-x);
  printf("%d", A + B + C + D);
  return 0;
}
```

答案:              J         (2 分)

全卷完