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	7 (A 至 L)	20
總分			**45**

甲部 (25 分)

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

1. 以下程序的輸出是？

Pascal 版本

```
var
  a, b: longint;
begin
  a := 8;
  b := 2016;
  while (b >= a) do
  begin
    b := b + 4;
    a := a * 2
  end;
  write(b)
end.
```

C 版本

```
#include <stdio.h>
int a, b;
int main() {
  a = 8;
  b = 2016;
  while (b >= a) {
    b = b + 4;
    a = a * 2;
  }
  printf("%d", b);
}
```

A. 2044
B. 2048
C. 2052
D. 2056

2. 以下程序的輸出是？

Pascal 版本

```
var
  sum: longint;
procedure abc(x, y: longint);
begin
  if x = 8 then
    sum := sum + y
  else
  begin
    abc(x + 1, y * 2);
    abc(x + 1, y * 2 + 1)
  end
end;
begin
  sum := 0;
  abc(0, 0);
  write(sum)
end.
```

C 版本

```
#include <stdio.h>
int sum;
void abc(int x, int y) {
  if (x == 8)
    sum = sum + y;
  else {
    abc(x + 1, y * 2);
    abc(x + 1, y * 2 + 1);
  }
}
int main() {
  sum = 0;
  abc(0, 0);
  printf("%d", sum);
}
```

A. 16320
B. 32640
C. 65408
D. 130816

3. 愛麗絲對著鮑伯這樣形容一棵樹:

- 此樹有 5 個節點。
- E 的父節點是 B。
- A 的父節點是 C。
- B 的父節點是 D。
- C 的父節點是 B。

以下哪項是樹根？

A. A
B. B
C. C
D. D

4. 設 T 是一棵有 45 個節點的樹。
定義一個節點的深度為該點與樹根之間的邊的數目。樹根的深度是 0。
在 T 中，深度為 i 的節點的數目是 $i + 1$。

以下哪一項是對的？

i. 如果葉節點的數目是 9，T 是一棵二元樹。
ii. 所有節點的深度的總和是 72。
iii. 葉節點的數目最多是 36。

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

5. 現有 n 個人 $P_1, P_2, ..., P_n$，而 n 是大於 2 的整數。所有人同時說了一句話：

P_1：有 1 個人在說謊。
P_2：有 2 個人在說謊。
...
P_i：有 i 個人在說謊。
...
P_{n-1}：有 $n-1$ 個人在說謊。
P_n：有 n 個人在說謊。

誰是唯一一個說了真話的人？

A. P_1
B. P_{n-1}
C. P_n
D. 以上皆非

6. 以下程序有多少個不同的可能輸出？假設輸入必定為 32 位元的帶符號整數。

Pascal 版本

```
var
  x: longint;
begin
  read(x);
  write((x mod 5) mod 3)
end.
```

C 版本

```
#include <stdio.h>
int x;
int main() {
  scanf("%d", &x);
  printf("%d", (x % 5) % 3);
  return 0;
}
```

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

7. 僅考慮以下排序演算法在最差情況下的表現，哪一個演算法擁有最佳的時間複雜度？

A. 合併排序法
B. 冒泡排序法
C. 快速排序法
D. 選擇排序法

8. 以下哪個數據結構可運行二分搜尋法？

i. 堆疊
ii. 隊列
iii. 鏈表

A. 只有 iii
B. 只有 i 和 ii
C. i, ii 和 iii
D. 以上皆非

9. 考慮以下程序：

Pascal 版本

```
var
  n, i: longint;
begin
  read(n);
  for i := 2 to n - 1 do
  begin
    if (i mod n = 0) then
    begin
      write('Composite');
      halt
    end
  end;
  write('Prime')
end.
```

C 版本

```
#include <stdio.h>
int n, i;
int main() {
  scanf("%d", &n);
  for (i = 2; i <= n - 1; i++) {
    if (i % n == 0) {
      printf("Composite");
      return 0;
    }
  }
  printf("Prime");
  return 0;
}
```

輸入 n 必定是個 2 和 100000000（包含 2 與 100000000）之間的整數，以下哪項敍述是正確的？

A. 雖然不是最有效率的方法，程序能正確判斷 n 是質數還是合成數
B. 程序可能沒有輸出
C. 存在一個會導致運行錯誤的合法輸入 n
D. 當 n 是個平方數，如 4, 9, 16, ...，程序必定會輸出 "Prime"

10. 因中英文版本有異，此題取消

對於 11 至 12 題，考慮以下程序：

Pascal 版本

```
var
  i, j: longint;
  tri: array[0..14, 0..14] of longint;
begin
  for i := 0 to 14 do
    for j := 0 to 14 do
      tri[i][j] := 0;
  tri[0][0] := 1;
  for i := 1 to 14 do
  begin
    tri[i][0] := 1;
    tri[i][i] := 1;
    for j := 1 to i - 1 do
      tri[i][j] := tri[i - 1][j] +
         tri[i - 1][j + 1]
  end;
  write(tri[7][1], ' ', tri[13][7])
end.
```

C 版本

```
#include <stdio.h>
int tri[15][15], i, j;
int main(){
  for (i = 0; i <= 14; i++)
    for (j = 0; j <= 14; j++)
      tri[i][j] = 0;
  tri[0][0] = 1;
  for (i = 1; i <= 14; i++) {
    tri[i][0] = 1;
    tri[i][i] = 1;
    for (j = 1; j <= i - 1; j++)
      tri[i][j] = tri[i - 1][j] +
         tri[i - 1][j + 1];
  }
  printf("%d %d", tri[7][1],
     tri[13][7]);
  return 0;
}
```

11. 第一個輸出的數是？
   A. 7
   B. 13
   C. 21
   D. 1716

12. 第二個輸出的數是？
   A. 7
   B. 13
   C. 21
   D. 1716

13. 考慮以下程序：

Pascal 版本

```
var
  n, i, sum: longint;
begin
  read(n);
  sum := 0;
  for i := 0 to n do
    sum := sum + (i and (n - i));
  write(sum)
end.
```

C 版本

```
#include <stdio.h>
int n, i, sum;
int main() {
  scanf("%d", &n);
  sum = 0;
  for (i = 0; i <= n; i++)
    sum = sum + (i & (n - i));
  printf("%d", sum);
  return 0;
}
```

下列哪個輸入 n 會使輸出最小?
   A. 31
   B. 32
   C. 33
   D. 34

14. 以下程序的輸出是？

Pascal 版本

```
var
  i, sum:  longint;
begin
  sum := 0;
  for i := 0 to 100 do
    sum := sum + (i xor 7);
  write(sum)
end.
```

C 版本

```
#include <stdio.h>
int i, sum;
int main(){
  sum = 0;
  for (i = 0; i <= 100; i++)
    sum = sum + (i ^ 7);
  printf("%d", sum);
  return 0;
}
```

A. 5049
B. 5050
C. 5065
D. 5066

15. 以下哪項可以用一個或以上的陣列實現？假設所用的陣列有足夠容量。

i. 堆疊
ii. 隊列
iii. 鏈表

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

16. 考慮以下程序：（變量宣告與 P_push, P_pop, Q_push, Q_pop 的實現已省略）

Pascal 版本

```
var
  i: longint;
  a: array[0..7] of longint;
begin
  for i := 0 to 7 do
    read(a[i]);
  for i := 0 to 7 do
    P_push(a[i]);
  for i := 0 to 7 do
    Q_push(P_pop());
  for i := 0 to 6 do
    write(Q_pop(), ' ');
  write(Q_pop())
end.
```

C 版本

```
#include <stdio.h>
int i, a[8];
int main() {
  for (i = 0; i <= 7; i++)
    scanf("%d", &a[i]);
  for (i = 0; i <= 7; i++)
    P_push(a[i]);
  for (i = 0; i <= 7; i++)
    Q_push(P_pop());
  for (i = 0; i <= 6; i++)
    printf("%d ", Q_pop());
  printf("%d", Q_pop());
}
```

設 P、Q 為數據結構。P_push()、Q_push()、P_pop() 及 Q_pop() 是對應該數據結構的程序／函數。輸入是一列以空格分開的 8 個整數。以下哪個 P、Q 的組合會使輸出為輸入的倒轉？

i. P: 堆疊, Q: 堆疊
ii. P: 堆疊, Q: 隊列
iii. P: 隊列, Q: 堆疊
iv. P: 隊列, Q: 隊列

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

對於 17 至 18 題，考慮以下四個程序。

Pascal 版本

```
procedure a();
var
  i, j: longint;
begin
  for i := 0 to 10 do
    for j := i + 1 downto 0 do
      write('*')
end;

procedure b();
var
  i, j: longint;
begin
  i := 20;
  while (i >= 0) do
  begin
    j := (i - 3) div 2;
    i := j;
    while (i > 0) do
    begin
      write('*');
      dec(i)
    end;
    dec(i)
  end
end;

procedure c();
var
  i, j, k: longint;
begin
  for i := 1 to 4 do
    for j := 1 to 4 do
      for k := 1 to 4 do
        if (i * j * k <= 20) then
          write('*')
end;

procedure d();
var
  i, j, k: longint;
begin
  for i := 1 to 6 do
    for j := 1 to 5 do
      for k := j to 7 do
        if ((i + j + k) mod 3 = 0) then
          write('*')
end;
```

C 版本

```
void a() {
  int i, j;
  for (i = 0; i <= 10; i++)
    for (j = i + 1; j >= 0; j--)
      printf("*");
}

void b() {
  int i, j;
  i = 20;
  while (i >= 0) {
    j = (i - 3) / 2;
    i = j;
    while (i > 0) {
      printf("*");
      i--;
    }
    i--;
  }
}

void c() {
  int i, j, k;
  for (i = 1; i <= 4; i++)
    for (j = 1; j <= 4; j++)
      for (k = 1; k <= 4; k++)
        if(i * j * k <= 20)
          printf("*");
}

void d() {
  int i, j, k;
  for (i = 1; i <= 6; i++)
    for (j = 1; j <= 5; j++)
      for (k = j; k <= 7; k++)
        if ((i + j + k) % 3 == 0)
          printf("*");
}
```

17. 哪個程序輸出最少的 "*"？

A. `a()`
B. `b()`
C. `c()`
D. `d()`

18. 哪個程序輸出最多的 "*"？

A. `a()`
B. `b()`
C. `c()`
D. `d()`

19. 以下程序的輸出是？

Pascal 版本

```
var
  i, j: longint;
  x: array[0..999] of longint;
begin
  for i := 1 to 999 do
    x[i] := 0;
  for i := 2 to 999 do
    if x[i] = 0 then
    begin
      j := i;
      while (j <= 999) do
      begin
        x[j] := x[j] + 1;
        j := j + i
      end
    end;
  write(x[30] + x[37] +
     x[60] + x[999])
end.
```

C 版本

```
#include <stdio.h>

int i, j, x[1000];

int main() {
  for (i = 1; i <= 999; i++)
    x[i] = 0;
  for (i = 2; i <= 999; i++)
    if (x[i] == 0) {
      j = i;
      while (j <= 999) {
        x[j] = x[j] + 1;
        j = j + i;
      }
    }
  printf("%d", x[30] + x[37] +
     x[60] + x[999]);
  return 0;
}
```

A. 8
B. 9
C. 10
D. 11

20. 考慮以下程序段，`sort(x, y)` 是一個能把 `a[x]` 至 `a[y]` (包含 `a[x]` 及 `a[y]`) 按不遞減次序排列的子程序。（子程序 `sort` 的實現已省略）

Pascal 版本

```
var
  a: array[0..14] of longint;
  i: longint;
begin
  for i := 0 to 14 do
    read(a[i]);
  sort(0, 3);
  sort(6, 10);
  sort(2, 9);
  sort(10, 14)
end.
```

C 版本

```
#include <stdio.h>
int a[15];
int i;
int main() {
  for (i = 0; i <= 14; i++)
    scanf("%d", &a[i]);
  sort(0, 3);
  sort(6, 10);
  sort(2, 9);
  sort(10, 14);
  return 0;
}
```

程序段完結時，以下哪些必定為真？

i. `a[9] >= a[0]`
ii. `a[7] >= a[0]`
iii. `a[14] >= a[5]`

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

21. 考慮以下程序：

Pascal 版本

```
var
  n, i, x, y: longint;
  a: array[0..9] of longint;
begin
  for i := 0 to 9 do
    read(a[i]);
  x := 0;
  y := 9;
  while (x <> y) do
  begin
    if (a[x] < a[y]) then
      inc(x)
    else
      dec(y)
  end;
  write(x)
end.
```

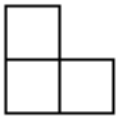
C 版本

```
#include <stdio.h>
int n, i, x, y;
int a[10];
int main() {
  for (i = 0; i <= 9; i++)
    scanf("%d", &a[i]);
  x = 0;
  y = 9;
  while (x != y) {
    if (a[x] < a[y])
      x++;
    else
      y--;
  }
  printf("%d", x);
  return 0;
}
```

以下哪組輸入能產生最大的輸出值？

A. 1 0 5 7 8 0 3 4 6 2
B. 7 1 3 9 4 8 2 5 6 0
C. 5 4 9 0 3 2 7 8 9 1
D. 1 2 7 4 3 0 8 6 5 6

22. L-塊是一個能夠以四個不同旋轉方式覆蓋三個格子的形狀，以下是一個 L-塊：



格網會被完全填充當且僅當每個格子都被一個 L-塊覆蓋。
假設現在我們要用 16 個 L-塊填充一個缺少了一個格子的 7×7 格網，以下哪一個格網能夠被完全填充？（你可以把 $(1,1)$ 及 $(7,7)$ 分別當為左上角與右下角的格子，當中 (x,y) 表示位於第 x 行第 y 列的格子）

i. 缺少了格子 $(7,7)$ 的格網
ii. 缺少了格子 $(3,6)$ 的格網
iii. 缺少了格子 $(2,5)$ 的格網

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

23. 考慮下列的不等方程組

$$\begin{cases} a \geq b - 1 \\ b \geq c + 2 \\ c \geq a - 3 \\ 0 \leq c \leq 3 \end{cases}$$

以上的不等方程組共有多少組整數解 (a,b,c)？

A. 8
B. 16
C. 24
D. 32

24. Skylake 為

A. 某一中央處理器版本的代號
B. 某一操作系統版本的代號
C. 某一編程語言的名稱
D. 某一伺服器軟件的名稱

25. 一開始時，你有 3 名村民，倉庫裡有 200 單位的食物。你可以消耗倉庫內的 50 單位的食物來即時生產一名村民。每名村民每秒生產 1 單位的食物，並在每秒完結時將食物放入倉庫。

求使得倉庫內有 500 單位的食物的最快時間。

A. 69
B. 70
C. 72
D. 100

甲部完

乙部 (20 分)

下列各空格分別命名為 A 至 L，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. 請完成以下程序，使其輸出 `"place"`。

Pascal 版本

```
var
  s: string;
  i: longint;
begin
  s := 'pace';
  for ____A____ do
    s[i] := ____B____;
  s[i] := 'l';
  setlength(s, 5);
  write(s)
end.
```

C 版本

```
#include <stdio.h>
char s[256] = "pace";
int i;
int main() {
  for (____A____)
    s[i] = ____B____;
  s[i] = 'l';
  s[5] = 0;
  printf("%s", s);
  return 0;
}
```

答案： ____A____ (1 分)
答案： ____B____ (1 分)

2. 考慮下列程序段：

Pascal 版本

```
function sq_cmp(a, b: longint): longint;
begin
  if (a * a > b * b) then
    sq_cmp := 1
  else
    sq_cmp := 0
end;
function equals(a, b: longint): longint;
begin
  equals := sq_cmp(____C____)
end;
function greater(a, b: longint): longint;
begin
  greater := sq_cmp(____D____)
end;
```

C 版本

```
int sq_cmp(int a, int b) {
  if (a * a > b * b)
    return 1;
  else
    return 0;
}
int equals(int a, int b) {
  return sq_cmp(____C____);
}
int greater(int a, int b) {
  return sq_cmp(____D____);
}
```

完成以上程序段, 使得對於在 -100 及 100 之間 (含 -100 及 100) 的整數 `a`、`b`，`equals(a, b)` 會在 `a == b` (**Pascal**：`a = b`) 時返回 `1`，否則返回 `0`，而 `greater(a, b)` 會在 `a > b` 時返回 `1`，否則返回 `0`。
你只可以使用 `a`、`b`、常數、逗號`","`、空格、括號`"()"` 以及下列運算符: `+`、`-`、`*`、`/` (**Pascal**：`div`)、`%` (**Pascal**：`mod`)。

答案： ____C____ (2 分)
答案： ____D____ (2 分)

3. 考慮以下程序段：

Pascal 版本

```
var
  n, ans: longint;
procedure f(a, b, n: longint);
begin
  if (n = 0) then
    ans := b - a
  else if (n mod 2 = 0) then
    f(a + 1, b, n div 2)
  else
    f(a, b + 1, n div 2)
end;
begin
  read(n);
  ans := 0;
  if ((n >= 0) and (n <= 2016)) then
    f(0, 0, n);
  write(ans)
end.
```

C 版本

```
#include <stdio.h>
int n, ans;
void f(int a, int b, int n) {
  if (n == 0)
    ans = b - a;
  else if (n % 2 == 0)
    f(a + 1, b, n / 2);
  else
    f(a, b + 1, n / 2);
}

int main() {
  scanf("%d", &n);
  ans = 0;
  if ((n >= 0) && (n <= 2016))
    f(0, 0, n);
  printf("%d", ans);
  return 0;
}
```

已知以上程序的輸出為 3，輸入的最小及最大可能值是甚麼? 輸入的最小可能值為____E____，最大可能值為____F____。

答案： _____E_____ (1 分)
答案： _____F_____ (2 分)

4. 給定陣列 a (**Pascal**：a[1..n] of longint, **C**：int a[n]) 內的整數均不同並已按升序排序，整數 n 儲存陣列的大小而整數 x 儲存要找出的數。以下程序試圖實現對分搜索（二分搜尋法）來判斷整數 x 是否存在於陣列 a 內。但是程序中有一錯誤，並只需更改一行便能修正，請找出並將其改正。假設變量 l，r，mid 已被宣告為 int (**Pascal**：longint)。

Pascal 版本

```
11 l := 1;
12 r := n;
13 while (l <= r) do
14 begin
15   mid := (l + r) div 2;
16   if (a[mid] < x) then
17     l := mid + 1
18   else
19     r := mid - 1
20 end;
21 if (a[mid] = x) then
22   write('Found')
23 else
24   write('Not found');
```

C 版本

```
51 l = 0;
52 r = n - 1;
53 while (l <= r)
54 {
55   mid = (l + r) / 2;
56   if (a[mid] < x)
57     l = mid + 1;
58   else
59     r = mid - 1;
60 }
61 if (a[mid] == x)
62   printf("Found");
63 else
64   printf("Not found");
```

行數： ____G____ (1 分)
改正： _____H_____ (2 分)

5. 以下程序試圖輸出正整數 x 的所有因數：

Pascal 版本

```
var
  i, k, t, x: longint;
begin
  read(x);
  k := x;
  writeln(1);
  if (x > 1) then
  begin
    for i := 2 to x do
    begin
      t := 1;
      while (k mod i = 0) do
      begin
        k := k div i;
        t := t * i;
        writeln(t)
      end
    end
  end
end.
```

C 版本

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

I1. 寫下一個大於 30 的正整數輸入 x，使得以上程序正確地輸出 x 的所有因數。
答案： I1

I2. 寫下一個大於 30 的正整數輸入 x，使得以上程序錯誤地輸出 x 的所有因數。
答案： I2

只有全對才能獲取 2 分。

6. 我們稱以下的圖形為「十字形」。

```
 *
***
 *
```

考慮以下含有兩個十字形的坐標方格，這兩個十字形的中心點分別位於坐標 $(2,2)$ 及 $(4,4)$。
（"." 代表空格，當中 (x,y) 表示位於第 x 行第 y 列的格子）

```
.*...
***..
.*.*.
..***
...*.
```

我們稱兩個十字形「互相接觸」若他們有共同角，但沒有共同邊。
我們稱兩個十字形「互相連繫」若他們有共同邊。

可見，以上兩個十字形是互相接觸的。

考慮以下程序段，(a,b) 儲存了第一個十字形的中心點的坐標，(x,y) 儲存了第二個十字形的中心點的坐標。

Pascal 版本

```
read(a, b, x, y);
if (____J____) then
  write('They are connected');
if (____K____) then
  write('They touch each other');
```

C 版本

```
scanf("%d%d%d%d", &a, &b, &x, &y);
if (____J____)
  printf("They are connected");
if (____K____)
  printf("They touch each other");
```

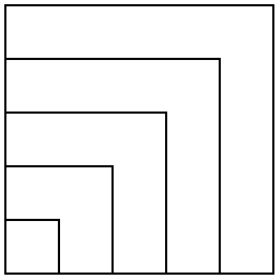
完成以下程序，令其能正確判斷兩個十字形是否互相接觸或互相連繫。
程序應輸出 "They touch each other" 當且僅當它們互相接觸。
程序應輸出 "They are connected" 當且僅當它們互相連繫。
兩個十字形保證不會重疊。
備註：你可以使用 abs 函數。

答案： _____J_____ (2 分)
答案： _____K_____ (2 分)

7. Logo（圖龜）是一種用來繪畫形狀的編程語言。開始的時候畫面上會有一隻「龜」指向上方，然後你可以用指令移動它，使它的路徑留在畫面上。舉個例說，考慮以下程序：

```
to sq :size
  if :size = 0 [ stop ]
  repeat 4 [ fd :size rt 90 ]
  sq :size - 50
end
sq 250
```

造出的圖案為：



現在，在答題紙上畫出以下程序造出的圖案。大小比例不拘，些微的誤差是可以接受的。
提示：repeat 為重覆，fd 為前進，lt 為左轉，rt 為右轉。

```
to shape :size
  if :size = 30 [ stop ]
  repeat 3 [ fd :size / 3 lt 60 shape :size / 3 rt 60 fd :size * 2 / 3 rt 120 ]
end
shape 270
```

答案： _____L_____ (2 分)

全卷完