

1. 右側程式正確的輸出應該如下：

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

在不修改右側程式之第 4 行及第 7 行程式碼的前提下，最少需修改幾行程式碼以得到正確輸出？

- (A) 1
- (B) 2
- (C) 3
- (D) 4

```
1  int k = 4;  
2  int m = 1;  
3  for (int i=1; i<=5; i=i+1) {  
4      for (int j=1; j<=k; j=j+1) {  
5          printf (" ");  
6      }  
7      for (int j=1; j<=m; j=j+1) {  
8          printf ("*");  
9      }  
10     printf ("\n");  
11     k = k - 1;  
12     m = m + 1;  
13 }
```

2. 給定一陣列  $a[10]=\{ 1, 3, 9, 2, 5, 8, 4, 9, 6, 7 \}$ ，i.e.,  $a[0]=1, a[1]=3, \dots, a[8]=6, a[9]=7$ ，以  $f(a, 10)$  呼叫執行右側函式後，回傳值為何？

- (A) 1
- (B) 2
- (C) 7
- (D) 9

```
int f (int a[], int n) {  
    int index = 0;  
    for (int i=1; i<=n-1; i=i+1) {  
        if (a[i] >= a[index]) {  
            index = i;  
        }  
    }  
    return index;  
}
```

3. 給定一整數陣列  $a[0]$ 、 $a[1]$ 、...、 $a[99]$  且  $a[k]=3k+1$ ，以  $value=100$  呼叫以下兩函式，假設函式 **f1** 及 **f2** 之 **while** 迴圈主體分別執行  $n1$  與  $n2$  次 (i.e, 計算 **if** 敘述執行次數，不包含 **else if** 敘述)，請問  $n1$  與  $n2$  之值為何？ 註：  $(low + high)/2$  只取整數部分。

```
int f1(int a[], int value) {
    int r_value = -1;
    int i = 0;
    while (i < 100) {
        if (a[i] == value) {
            r_value = i;
            break;
        }
        i = i + 1;
    }
    return r_value;
}
```

```
int f2(int a[], int value) {
    int r_value = -1;
    int low = 0, high = 99;
    int mid;
    while (low <= high) {
        mid = (low + high)/2;
        if (a[mid] == value) {
            r_value = mid;
            break;
        }
        else if (a[mid] < value) {
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return r_value;
}
```

- (A)  $n1=33, n2=4$   
(B)  $n1=33, n2=5$   
(C)  $n1=34, n2=4$   
(D)  $n1=34, n2=5$

4. 經過運算後，右側程式的輸出為何？

- (A) 1275  
(B) 20  
(C) 1000  
(D) 810

```
for (i=1; i<=100; i=i+1) {
    b[i] = i;
}
a[0] = 0;
for (i=1; i<=100; i=i+1) {
    a[i] = b[i] + a[i-1];
}
printf ("%d\n", a[50]-a[30]);
```

5. 函數 **f** 定義如下，如果呼叫 **f(1000)**，指令 **sum=sum+i** 被執行的次數最接近下列何者？

- (A) 1000
- (B) 3000
- (C) 5000
- (D) 10000

```
int f (int n) {
    int sum=0;
    if (n<2) {
        return 0;
    }
    for (int i=1; i<=n; i=i+1) {
        sum = sum + i;
    }
    sum = sum + f(2*n/3);
    return sum;
}
```

6. List 是一個陣列，裡面的元素是 element，它的定義如右。List 中的每一個 element 利用 **next** 這個整數變數來記錄下一個 element 在陣列中的位置，如果沒有下一個 element，**next** 就會記錄 -1。所有的 element 串成了一個串列 (linked list)。例如在 **list** 中有三筆資料

1	2	3
data = 'a'	data = 'b'	data = 'c'
next = 2	next = -1	next = 1

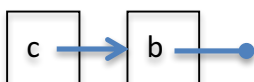
它所代表的串列如下圖



```
struct element {
    char data;
    int next;
}

void RemoveNextElement (element
list[], int current) {
    if (list[current].next != -1) {
        /*移除 current 的下一個 element*/
        
    }
}
```

**RemoveNextElement** 是一個程序，用來移除串列中 **current** 所指向的下一個元素，但是必須保持原始串列的順序。例如，若 **current** 為 3 (對應到 **list[3]**)，呼叫完 **RemoveNextElement** 後，串列應為



請問在空格中應該填入的程式碼為何？

- (A) **list[current].next = current ;**
- (B) **list[current].next = list[list[current].next].next ;**
- (C) **current = list[list[current].next].next ;**
- (D) **list[list[current].next].next = list[current].next ;**

7. 請問以 **a(13,15)** 呼叫右側 **a()** 函式，函式執行完後其回傳值為何？

- (A) 90
- (B) 103
- (C) 93
- (D) 60

```
int a(int n, int m) {  
    if (n < 10) {  
        if (m < 10) {  
            return n + m ;  
        }  
        else {  
            return a(n, m-2) + m ;  
        }  
    }  
    else {  
        return a(n-1, m) + n ;  
    }  
}
```

8. 一個費式數列定義第一個數為 0 第二個數為 1 之後的每個數都等於前兩個數相加，如下所示：  
0、1、1、2、3、5、8、13、21、34、55、89...。  
右列的程式用以計算第 N 個( $N \geq 2$ )費式數列的數值，  
請問 (a) 與 (b) 兩個空格的敘述(statement)應該為何？

- (A) (a) **f[i]=f[i-1]+f[i-2]**      (b) **f[N]**
- (B) (a) **a = a + b**      (b) **a**
- (C) (a) **b = a + b**      (b) **b**
- (D) (a) **f[i]=f[i-1]+f[i-2]**      (b) **f[i]**

```
int a=0;  
int b=1;  
int i, temp, N;  
...  
for (i=2; i<=N; i=i+1) {  
    temp = b;  
    _____ (a) _____ ;  
    a = temp;  
    printf ("%d\n", _____ (b) _____ );  
}
```

9. 請問右側程式輸出為何？

- (A) 1
- (B) 4
- (C) 3
- (D) 33

```
int A[5], B[5], i, c;
...
for (i=1; i<=4; i=i+1) {
    A[i] = 2 + i*4;
    B[i] = i*5;
}
c = 0;
for (i=1; i<=4; i=i+1) {
    if (B[i] > A[i]) {
        c = c + (B[i] % A[i]);
    }
    else {
        c = 1;
    }
}
printf ("%d\n", c);
```

10. 給定右側 **g()** 函式，**g(13)** 回傳值為何？

- (A) 16
- (B) 18
- (C) 19
- (D) 22

```
int g(int a) {
    if (a > 1) {
        return g(a - 2) + 3;
    }
    return a;
}
```

11. 定義 **a[n]** 為一陣列(array)，陣列元素的指標為 0 至 n-1。若要將陣列中 **a[0]** 的元素移到 **a[n-1]**，右側程式片段空白處該填入何運算式？

- (A) **n+1**
- (B) **n**
- (C) **n-1**
- (D) **n-2**

```
int i, hold, n;
...
for (i=0; i<=____; i=i+1) {
    hold = a[i];
    a[i] = a[i+1];
    a[i+1] = hold;
}
```

12. 給定右側函式 **f1()** 及 **f2()**。**f1(1)** 運算過程中，以下敘述何者為錯？

- (A) 印出的數字最大的是 4
- (B) **f1** 一共被呼叫二次
- (C) **f2** 一共被呼叫三次
- (D) 數字 2 被印出兩次

```
void f1 (int m) {
    if (m > 3) {
        printf ("%d\n", m);
        return;
    }
    else {
        printf ("%d\n", m);
        f2(m+2);
        printf ("%d\n", m);
    }
}

void f2 (int n) {
    if (n > 3) {
        printf ("%d\n", n);
        return;
    }
    else {
        printf ("%d\n", n);
        f1(n-1);
        printf ("%d\n", n);
    }
}
```

13. 右側程式片段擬以輾轉除法求 **i** 與 **j** 的最大公因數。請問 **while** 迴圈內容何者正確？

- (A) **k = i % j;**  
**i = j;**  
**j = k;**
- (B) **i = j;**  
**j = k;**  
**k = i % j;**
- (C) **i = j;**  
**j = i % k;**  
**k = i;**
- (D) **k = i;**  
**i = j;**  
**j = i % k;**

```
i = 76;
j = 48;
while ((i % j) != 0) {
    _____
    _____
    _____
}
printf ("%d\n", j);
```

14. 右側程式輸出為何？

- (A) bar: 6  
bar: 1  
bar: 8
- (B) bar: 6  
foo: 1  
bar: 3
- (C) bar: 1  
foo: 1  
bar: 8
- (D) bar: 6  
foo: 1  
foo: 3

```
void foo (int i) {  
    if (i <= 5) {  
        printf ("foo: %d\n", i);  
    }  
    else {  
        bar(i - 10);  
    }  
}  
  
void bar (int i) {  
    if (i <= 10) {  
        printf ("bar: %d\n", i);  
    }  
    else {  
        foo(i - 5);  
    }  
}  
  
void main() {  
    foo(15106);  
    bar(3091);  
    foo(6693);  
}
```

15. 若以 **f(22)** 呼叫右側 **f()** 函式，總共會印出多少數字？

- (A) 16
- (B) 22
- (C) 11
- (D) 15

```
void f(int n) {  
    printf ("%d\n", n);  
    while (n != 1) {  
        if ((n%2)==1) {  
            n = 3*n + 1;  
        }  
        else {  
            n = n / 2;  
        }  
        printf ("%d\n", n);  
    }  
}
```

16. 右側程式執行過後所輸出數值為何？

- (A) 11
- (B) 13
- (C) 15
- (D) 16

```
void main () {  
    int count = 10;  
    if (count > 0) {  
        count = 11;  
    }  
    if (count > 10) {  
        count = 12;  
        if (count % 3 == 4) {  
            count = 1;  
        }  
        else {  
            count = 0;  
        }  
    }  
    else if (count > 11) {  
        count = 13;  
    }  
    else {  
        count = 14;  
    }  
    if (count) {  
        count = 15;  
    }  
    else {  
        count = 16;  
    }  
  
    printf ("%d\n", count);  
}
```



17. 右側程式片段主要功能為：輸入六個整數，檢測並印出最後一個數字是否為六個數字中最小的值。然而，這個程式是錯誤的。請問以下哪一組測試資料可以測試出程式有誤？

- (A) 11 12 13 14 15 3  
(B) 11 12 13 14 25 20  
(C) 23 15 18 20 11 12  
(D) 18 17 19 24 15 16

```
#define TRUE 1
#define FALSE 0
int d[6], val, allBig;

...
for (int i=1; i<=5; i=i+1) {
    scanf ("%d", &d[i]);
}
scanf ("%d", &val);
allBig = TRUE;
for (int i=1; i<=5; i=i+1) {
    if (d[i] > val) {
        allBig = TRUE;
    }
    else {
        allBig = FALSE;
    }
}
if (allBig == TRUE) {
    printf ("%d is the smallest.\n", val);
}
else {
    printf ("%d is not the smallest.\n", val);
}
}
```

18. 程式編譯器可以發現下列哪種錯誤？

- (A) 語法錯誤  
(B) 語意錯誤  
(C) 邏輯錯誤  
(D) 以上皆是

19. 大部分程式語言都是以列為主的方式儲存陣列。在一個 8x4 的陣列(array) **A** 裡，若每個元素需要兩單位的記憶體大小，且若 **A[0][0]** 的記憶體位址為 108 (十進制表示)，則 **A[1][2]** 的記憶體位址為何？

- (A) 120  
(B) 124  
(C) 128  
(D) 以上皆非

20. 右側為一個計算  $n$  階層的函式，請問該如何修改才會得到正確的結果？

```
1. int fun (int n) {  
2.     int fac = 1;  
3.     if (n >= 0) {  
4.         fac = n * fun(n - 1);  
5.     }  
6.     return fac;  
7. }
```

- (A) 第 2 行，改為 `int fac = n;`
- (B) 第 3 行，改為 `if (n > 0) {`
- (C) 第 4 行，改為 `fac = n * fun(n+1);`
- (D) 第 4 行，改為 `fac = fac * fun(n-1);`

21. 右側程式碼，執行時的輸出為何？

- (A) 0 2 4 6 8 10
- (B) 0 1 2 3 4 5 6 7 8 9 10
- (C) 0 1 3 5 7 9
- (D) 0 1 3 5 7 9 11

```
void main() {  
    for (int i=0; i<=10; i=i+1) {  
        printf ("%d ", i);  
        i = i + 1;  
    }  
    printf ("\n");  
}
```

22. 右側 `f()` 函式執行後所回傳的值為何？

- (A) 1023
- (B) 1024
- (C) 2047
- (D) 2048

```
int f() {  
    int p = 2;  
    while (p < 2000) {  
        p = 2 * p;  
    }  
    return p;  
}
```

23. 右側 **f()** 函式 (a), (b), (c) 處需分別填入哪些數字，方能使得 **f(4)** 輸出 2468 的結果？

- (A) 1, 2, 1
- (B) 0, 1, 2
- (C) 0, 2, 1
- (D) 1, 1, 1

```
int f(int n) {
    int p = 0;
    int i = n;
    while (i >= ____ (a) ____ ) {
        p = 10 - ____ (b) ____ * i;
        printf ("%d", p);
        i = i - ____ (c) ____ ;
    }
}
```

24. 右側 **g(4)** 函式呼叫執行後，回傳值為何？

- (A) 6
- (B) 11
- (C) 13
- (D) 14

```
int f (int n) {
    if (n > 3) {
        return 1;
    }
    else if (n == 2) {
        return (3 + f(n+1));
    }
    else {
        return (1 + f(n+1));
    }
}

int g(int n) {
    int j = 0;
    for (int i=1; i<=n-1; i=i+1) {
        j = j + f(i);
    }
    return j;
}
```

25. 右側 **Mystery()** 函式 **else** 部分運算式應為何，才能使得 **Mystery(9)** 的回傳值為 34。

- (A) **x + Mystery(x-1)**
- (B) **x \* Mystery(x-1)**
- (C) **Mystery(x-2) + Mystery(x+2)**
- (D) **Mystery(x-2) + Mystery(x-1)**

```
int Mystery (int x) {
    if (x <= 1) {
        return x;
    }
    else {
        return _____ ;
    }
}
```

## 第 1 題 成績指標

### 問題描述

一次考試中，於所有及格學生中獲取最低分數者最為幸運，反之，於所有不及格同學中，獲取最高分數者，可以說是最為不幸，而此二種分數，可以視為成績指標。

請你設計一支程式，讀入全班成績(人數不固定)，請對所有分數進行排序，並分別找出不及格中最高分數，以及及格中最低分數。

當找不到最低及格分數，表示對於本次考試而言，這是一個不幸之班級，此時請你印出：「worst case」；反之，當找不到最高不及格分數時，請你印出「best case」。  
註：假設及格分數為 60，每筆測資皆為 0~100 間整數，且筆數未定。

### 輸入格式

第一行輸入學生人數，第二行為各學生分數(0~100 間)，分數與分數之間以一個空白間格。每一筆測資的學生人數為 1~20 的整數。

### 輸出格式

每筆測資輸出三行。

第一行由小而大印出所有成績，兩數字之間以一個空白間格，最後一個數字後無空白；

第二行印出最高不及格分數，如果全數及格時，於此行印出 best case；

第三行印出最低及格分數，當全數不及格時，於此行印出 worst case。

### 範例一：輸入

```
10
0 11 22 33 55 66 77 99 88 44
```

### 範例一：正確輸出

```
0 11 22 33 44 55 66 77 88 99
55
66
```

(說明) 不及格分數最高為 55，及格分數最低為 66。

### 範例二：輸入

```
1
13
```

### 範例二：正確輸出

```
13
13
worst case
```

(說明) 由於找不到最低及格分，因此第三行須印出「worst case」。

**範例三：輸入**

```
2
73 65
```

**範例三：正確輸出**

```
65 73
best case
65
```

（說明）由於找不到不及格分，因此第二行須印出「best case」。

**評分說明**

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。

## 第 2 題 矩陣轉換

### 問題描述

矩陣是將一群元素整齊的排列成一個矩形，在矩陣中的橫排稱為列 (row)，直排稱為行 (column)，其中以  $X_{ij}$  來表示矩陣  $X$  中的第  $i$  列第  $j$  行的元素。如圖一中， $X_{32} = 6$ 。

我們可以對矩陣定義兩種操作如下：

翻轉：即第一列與最後一列交換、第二列與倒數第二列交換、...依此類推。

旋轉：將矩陣以順時針方向轉 90 度。

例如：矩陣  $X$  翻轉後可得到  $Y$ ，將矩陣  $Y$  再旋轉後可得到  $Z$ 。

$X$		$Y$		$Z$		
1	4	3	6	1	2	3
2	5	2	5	4	5	6
3	6	1	4			

圖一

一個矩陣  $A$  可以經過一連串的旋轉與翻轉操作後，轉換成新矩陣  $B$ 。如圖二中， $A$  經過翻轉與兩次旋轉後，可以得到  $B$ 。給定矩陣  $B$  和一連串的操作，請算出原始的矩陣  $A$ 。

$A$		翻轉			旋轉			旋轉	$B$	
1	1	→	2	1	→	1	1	2	1	1
1	3		1	3		1	3	1	3	1
2	1		1	1					1	2

圖二

### 輸入格式

第一行有三個介於 1 與 10 之間的正整數  $R, C, M$ 。接下來有  $R$  行(line)是矩陣  $B$  的內容，每一行(line)都包含  $C$  個正整數，其中的第  $i$  行第  $j$  個數字代表矩陣  $B_{ij}$  的值。在矩陣內容後的一行有  $M$  個整數，表示對矩陣  $A$  進行的操作。第  $k$  個整數  $m_k$  代表第  $k$  個操作，如果  $m_k = 0$  則代表旋轉， $m_k = 1$  代表翻轉。同一行的數字之間都是以一個空白間格，且矩陣內容為 0~9 的整數。

### 輸出格式

輸出包含兩個部分。第一個部分有一行，包含兩個正整數  $R'$  和  $C'$ ，以一個空白

隔開，分別代表矩陣  $A$  的列數和行數。接下來有  $R'$  行，每一行都包含  $C'$  個正整數，且每一行的整數之間以一個空白隔開，其中第  $i$  行的第  $j$  個數字代表矩陣  $A_{ij}$  的值。每一行的最後一個數字後並無空白。

**範例一：輸入**

```
3 2 3
1 1
3 1
1 2
1 0 0
```

**範例一：正確輸出**

```
3 2
1 1
1 3
2 1
```

**(說明)**

如圖二所示

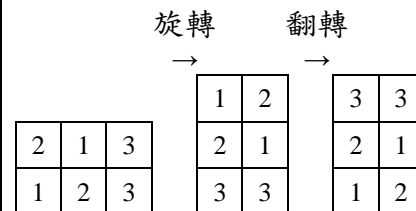
**範例二：輸入**

```
3 2 2
3 3
2 1
1 2
0 1
```

**範例二：正確輸出**

```
2 3
2 1 3
1 2 3
```

**(說明)**



**評分說明**

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。其中：

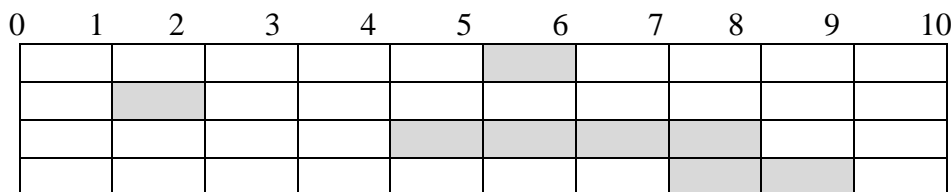
第一子題組共 30 分，其每個操作都是翻轉。

第二子題組共 70 分，操作有翻轉也有旋轉。

### 第 3 題 線段覆蓋長度

#### 問題描述

給定一維座標上一些線段，求這些線段所覆蓋的長度，注意，重疊的部分只能算一次。例如給定三個線段：(5, 6)、(1, 2)、(4, 8)、和(7, 9)，如下圖，線段覆蓋長度為 6。



#### 輸入格式：

第一列是一個正整數  $N$ ，表示此測試案例有  $N$  個線段。

接著的  $N$  列每一列是一個線段的開始端點座標和結束端點座標整數值，開始端點座標值小於等於結束端點座標值，兩者之間以一個空格區隔。

#### 輸出格式：

輸出其總覆蓋的長度。

#### 範例一：輸入

輸入	說明
5	此測試案例有 5 個線段
160 180	開始端點座標值與結束端點座標
150 200	開始端點座標值與結束端點座標
280 300	開始端點座標值與結束端點座標
300 330	開始端點座標值與結束端點座標
190 210	開始端點座標值與結束端點座標

#### 範例一：輸出

輸出	說明
110	測試案例的結果

#### 範例二：輸入

輸入	說明
1	此測試案例有 1 個線段
120 120	開始端點座標值與結束端點座標值

#### 範例二：輸出

輸出	說明
0	測試案例的結果



### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。每一個端點座標是一個介於 0~M 之間的整數，每筆測試案例線段個數上限為 N。其中：

第一子題組共 30 分， $M < 1000$ ， $N < 100$ ，線段沒有重疊。

第二子題組共 40 分， $M < 1000$ ， $N < 100$ ，線段可能重疊。

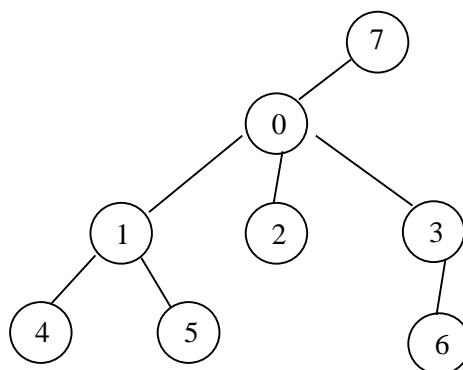
第三子題組共 30 分， $M < 100000000$ ， $N < 10000$ ，線段可能重疊。

## 第 4 題 血緣關係

### 問題描述

小宇有一個大家族。有一天，他發現記錄整個家族成員和成員間血緣關係的家族族譜。小宇對於最遠的血緣關係（我們稱之為"血緣距離"）有多遠感到很好奇。

右圖為家族的關係圖。0 是 7 的孩子，1、2 和 3 是 0 的孩子，4 和 5 是 1 的孩子，6 是 3 的孩子。我們可以輕易的發現最遠的親戚關係為 4(或 5)和 6，他們的"血緣距離"是 4(4~1, 1~0, 0~3, 3~6)。



給予任一家族的關係圖，請找出最遠的"血緣距離"。你可以假設只有一個人是整個家族成員的祖先，而且沒有兩個成員有同樣的小孩。

### 輸入格式

第一行為一個正整數  $n$  代表成員的個數，每人以  $0 \sim n-1$  之間惟一的編號代表。接著的  $n-1$  行，每行有兩個以一個空白隔開的整數  $a$  與  $b$  ( $0 \leq a, b \leq n-1$ )，代表  $b$  是  $a$  的孩子。

### 輸出格式

每筆測資輸出一行最遠"血緣距離"的答案。

#### 範例一：輸入

```

8
0 1
0 2
0 3
7 0
1 4
1 5
3 6
  
```

#### 範例一：正確輸出

4

#### (說明)

如題目所附之圖，最遠路徑為 4->1->0->3->6 或 5->1->0->3->6，距離為 4。

#### 範例二：輸入

```

4
0 1
0 2
2 3
  
```

#### 範例二：正確輸出

3

#### (說明)

最遠路徑為 1->0->2->3，距離為 3。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 3 秒，依正確通過測資筆數給分。其中，

第 1 子題組共 10 分，整個家族的祖先最多 2 個小孩，其他成員最多一個小孩， $2 \leq n \leq 100$ 。

第 2 子題組共 30 分， $2 \leq n \leq 100$ 。

第 3 子題組共 30 分， $101 \leq n \leq 2,000$ 。

第 4 子題組共 30 分， $1,001 \leq n \leq 100,000$ 。

1. 右側 F() 函式執行後，輸出為何？

- (A) 1 2
- (B) 1 3
- (C) 3 2
- (D) 3 3

```
void F( ) {  
    char t, item[] = {'2', '8', '3', '1', '9'};  
    int a, b, c, count = 5;  
    for (a=0; a<count-1; a=a+1) {  
        c = a;  
        t = item[a];  
        for (b=a+1; b<count; b=b+1) {  
            if (item[b] < t) {  
                c = b;  
                t = item[b];  
            }  
            if ((a==2) && (b==3)) {  
                printf ("%c %d\n", t, c);  
            }  
        }  
    }  
}
```

2. 右側 switch 敘述程式碼可以如何以 if-else 改寫？

- (A) if (x==10) y = 'a';  
 if (x==20 || x==30) y = 'b';  
 y = 'c';
- (B) if (x==10) y = 'a';  
 else if (x==20 || x==30) y = 'b';  
 else y = 'c';
- (C) if (x==10) y = 'a';  
 if (x>=20 && x<=30) y = 'b';  
 y = 'c';
- (D) if (x==10) y = 'a';  
 else if(x>=20 && x<=30) y = 'b';  
 else y = 'c';

```
switch (x) {  
    case 10: y = 'a'; break;  
    case 20:  
    case 30: y = 'b'; break;  
    default: y = 'c';  
}
```

3. 給定右側 G(), K() 兩函式，執行 G(3) 後所回傳的值為何？

- (A) 5  
(B) 12  
(C) 14  
(D) 15

```
int K(int a[], int n) {  
    if (n >= 0)  
        return (K(a, n-1) + a[n]);  
    else  
        return 0;  
}  
  
int G(int n){  
    int a[] = {5,4,3,2,1};  
    return K(a, n);  
}
```

4. 右側程式碼執行後輸出結果為何？

- (A) 3  
(B) 4  
(C) 5  
(D) 6

```
int a=2, b=3;  
int c=4, d=5;  
int val;  
  
val = b/a + c/b + d/b;  
printf ("%d\n", val);
```

5. 右側程式碼執行後輸出結果為何？

- (A) 2 4 6 8 9 7 5 3 1 9  
(B) 1 3 5 7 9 2 4 6 8 9  
(C) 1 2 3 4 5 6 7 8 9 9  
(D) 2 4 6 8 5 1 3 7 9 9

```
int a[9] = {1, 3, 5, 7, 9, 8, 6, 4, 2};  
int n=9, tmp;  
  
for (int i=0; i<n; i=i+1) {  
    tmp = a[i];  
    a[i] = a[n-i-1];  
    a[n-i-1] = tmp;  
}  
for (int i=0; i<=n/2; i=i+1)  
    printf ("%d %d ", a[i], a[n-i-1]);
```

6. 右側函式以 F(7) 呼叫後回傳值為 12，則 <condition> 應為何？

- (A)  $a < 3$   
(B)  $a < 2$   
(C)  $a < 1$   
(D)  $a < 0$

```
int F(int a) {  
    if ( <condition> )  
        return 1;  
    else  
        return F(a-2) + F(a-3);  
}
```

7. 若  $n$  為正整數，右側程式三個迴圈執行完畢後  $a$  值將為何？

- (A)  $n(n+1)/2$
- (B)  $n^3/2$
- (C)  $n(n-1)/2$
- (D)  $n^2(n+1)/2$

```
int a=0, n;  
...  
for (int i=1; i<=n; i=i+1)  
    for (int j=i; j<=n; j=j+1)  
        for (int k=1; k<=n; k=k+1)  
            a = a + 1;
```

8. 下面哪組資料若依序存入陣列中，將無法直接使用二分搜尋法搜尋資料？

- (A) a, e, i, o, u
- (B) 3, 1, 4, 5, 9
- (C) 10000, 0, -10000
- (D) 1, 10, 10, 10, 100

9. 右側是依據分數  $s$  評定等第的程式碼片段，正確的等第公式應為：

- 90~100 判為 A 等
- 80~89 判為 B 等
- 70~79 判為 C 等
- 60~69 判為 D 等
- 0~59 判為 F 等

這段程式碼在處理 0~100 的分數時，有幾個分數的等第是錯的？

- (A) 20
- (B) 11
- (C) 2
- (D) 10

```
if (s>=90) {  
    printf ("A \n");  
}  
else if (s>=80) {  
    printf ("B \n");  
}  
else if (s>60) {  
    printf ("D \n");  
}  
else if (s>70) {  
    printf ("C \n");  
}  
else {  
    printf ("F\n");  
}
```

10. 右側主程式執行完三次 G() 的呼叫後，p 陣列中有幾個元素的值為 0？

- (A) 1
- (B) 2
- (C) 3
- (D) 4

```
int K (int p[], int v) {  
    if (p[v]!=v) {  
        p[v] = K(p, p[v]);  
    }  
    return p[v];  
}  
  
void G (int p[], int l, int r) {  
    int a=K(p, l), b=K(p, r);  
    if (a!=b) {  
        p[b] = a;  
    }  
}  
  
int main (void) {  
    int p[5]={0, 1, 2, 3, 4};  
    G(p, 0, 1);  
    G(p, 2, 4);  
    G(p, 0, 4);  
    return 0;  
}
```

11. 右側程式片段執行後，count 的值為何？

- (A) 36
- (B) 20
- (C) 12
- (D) 3

```
int maze[5][5]= {{1, 1, 1, 1, 1},  
                 {1, 0, 1, 0, 1},  
                 {1, 1, 0, 0, 1},  
                 {1, 0, 0, 1, 1},  
                 {1, 1, 1, 1, 1}};  
  
int count=0;  
for (int i=1; i<=3; i=i+1) {  
    for (int j=1; j<=3; j=j+1) {  
        int dir[4][2] = {{-1,0}, {0,1}, {1,0}, {0,-1}};  
        for (int d=0; d<4; d=d+1) {  
            if (maze[i+dir[d][0]][j+dir[d][1]]==1) {  
                count = count + 1;  
            }  
        }  
    }  
}
```

12. 右側程式片段執行過程中的輸出為何？

- (A) 5 10 15 20
- (B) 5 11 17 23
- (C) 6 12 18 24
- (D) 6 11 17 22

```
int a = 5;
...
for (int i=0; i<20; i=i+1){
    i = i + a;
    printf ("%d ", i);
}
```

13. 若宣告一個字元陣列 `char str[20] = "Hello world!"`; 該陣列 `str[12]` 值為何？

- (A) 未宣告
- (B) `\0`
- (C) `!`
- (D) `\n`

14. 假設 `x, y, z` 為布林(boolean)變數，且 `x=TRUE`, `y=TRUE`, `z=FALSE`。請問下面各布林運算式的真假值依序為何？(TRUE 表真，FALSE 表假)

- `!(y || z) || x`
- `!y || (z || !x)`
- `z || (x && (y || z))`
- `(x || x) && z`

- (A) TRUE FALSE TRUE FALSE
- (B) FALSE FALSE TRUE FALSE
- (C) FALSE TRUE TRUE FALSE
- (D) TRUE TRUE FALSE TRUE



15. 右側程式片段執行過程的輸出為何？

- (A) 44  
(B) 52  
(C) 54  
(D) 63

```
int i, sum, arr[10];

for (int i=0; i<10; i=i+1)
    arr[i] = i;

sum = 0;
for (int i=1; i<9; i=i+1)
    sum = sum - arr[i-1] + arr[i] + arr[i+1];
printf ("%d", sum);
```

16. 右列程式片段中，假設 a, a\_ptr 和 a\_ptrptr 這三個變數都被正確宣告，且呼叫 G() 函式時的參數為 a\_ptr 及 a\_ptrptr。G() 函式的兩個參數型態該如何宣告？

- (A) (a) \*int, (b) \*int  
(B) (a) \*int, (b) \*\*int  
(C) (a) int\*, (b) int\*  
(D) (a) int\*, (b) int\*\*

```
void G ( __ (a) __ a_ptr, __ (b) __ a_ptrptr) {
    ...
}

void main () {
    int a = 1;
    // 加入 a_ptr, a_ptrptr 變數的宣告
    ...
    a_ptr = &a;
    a_ptrptr = &a_ptr;
    G (a_ptr, a_ptrptr);
}
```

17. 右側程式片段中執行後若要印出下列圖案，(a) 的條件判斷式該如何設定？

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

- (A) k > 2  
(B) k > 1  
(C) k > 0  
(D) k > -1

```
for (int i=0; i<=3; i=i+1) {
    for (int j=0; j<i; j=j+1)
        printf(" ");
    for (int k=6-2*i; __ (a) __ ; k=k-1)
        printf("*");
    printf("\n");
}
```

18. 給定右側 G() 函式，執行 G(1) 後所輸出的值為何？

- (A) 1 2 3
- (B) 1 2 3 2 1
- (C) 1 2 3 3 2 1
- (D) 以上皆非

```
void G (int a){  
    printf ("%d ", a);  
    if (a>=3)  
        return;  
    else  
        G(a+1);  
    printf ("%d ", a);  
}
```

19. 下列程式碼是自動計算找零程式的一部分，程式碼中三個主要變數分別為 Total (購買總額)，Paid (實際支付金額)，Change (找零金額)。但是此程式片段有冗餘的程式碼，請找出冗餘程式碼的區塊。

- (A) 冗餘程式碼在 A 區
- (B) 冗餘程式碼在 B 區
- (C) 冗餘程式碼在 C 區
- (D) 冗餘程式碼在 D 區

```
int Total, Paid, Change;  
...  
Change = Paid - Total;  
printf ("500 : %d pieces\n", (Change-Change%500)/500);  
Change = Change % 500;  
  
printf ("100 : %d coins\n", (Change-Change%100)/100);  
Change = Change % 100;  
  
// A 區  
printf ("50 : %d coins\n", (Change-Change%50)/50);  
Change = Change % 50;  
  
// B 區  
printf ("10 : %d coins\n", (Change-Change%10)/10);  
Change = Change % 10;  
  
// C 區  
printf ("5 : %d coins\n", (Change-Change%5)/5);  
Change = Change % 5;  
  
// D 區  
printf ("1 : %d coins\n", (Change-Change%1)/1);  
Change = Change % 1;
```

20. 右側程式執行後輸出為何？

- (A) 0
- (B) 10
- (C) 25
- (D) 50

```
int G (int B) {
    B = B * B;
    return B;
}

int main () {
    int A=0, m=5;

    A = G(m);
    if (m < 10)
        A = G(m) + A;
    else
        A = G(m);

    printf ("%d \n", A);
    return 0;
}
```

21. 右側 G() 應為一支遞迴函式，已知當 a 固定為 2，不同的變數 x 值會有不同的回傳值如下表所示。請找出 G() 函式中 (a) 處的計算式該為何？

```
int G (int a, int x) {
    if (x == 0)
        return 1;
    else
        return ____ (a) ____ ;
}
```

a 值	x 值	G(a, x) 回傳值
2	0	1
2	1	6
2	2	36
2	3	216
2	4	1296
2	5	7776

- (A)  $((2*a)+2) * G(a, x - 1)$
- (B)  $(a+5) * G(a-1, x - 1)$
- (C)  $((3*a)-1) * G(a, x - 1)$
- (D)  $(a+6) * G(a, x - 1)$

22. 如果  $X_n$  代表  $X$  這個數字是  $n$  進位，請問  $D02A_{16} + 5487_{10}$  等於多少？

- (A) 1100 0101 1001 1001<sub>2</sub>
- (B) 162631<sub>8</sub>
- (C) 58787<sub>16</sub>
- (D) F599<sub>16</sub>

23. 請問右側程式，執行完後輸出為何？

- (A) 2417851639229258349412352 7
- (B) 68921 43
- (C) 65537 65539
- (D) 134217728 6

```
int i=2, x=3;
int N=65536;

while (i <= N) {
    i = i * i * i;
    x = x + 1;
}
printf ("%d %d \n", i, x);
```

24. 右側  $G()$  為遞迴函式， $G(3, 7)$  執行後回傳值為何？

- (A) 128
- (B) 2187
- (C) 6561
- (D) 1024

```
int G (int a, int x) {
    if (x == 0)
        return 1;
    else
        return (a * G(a, x - 1));
}
```

25. 右側函式若以  $\text{search}(1, 10, 3)$  呼叫時， $\text{search}$  函式總共會被執行幾次？

- (A) 2
- (B) 3
- (C) 4
- (D) 5

```
void search (int x, int y, int z) {
    if (x < y) {
        t = ceiling ((x + y)/2);
        if (z >= t)
            search(t, y, z);
        else
            search(x, t - 1, z);
    }
}
```

註：ceiling() 為無條件進位至整數位。例如 ceiling(3.1)=4, ceiling(3.9)=4。

1. 給定一個 1x8 的陣列 **A**，**A = {0, 2, 4, 6, 8, 10, 12, 14}**。右側函式

**Search(x)** 真正目的是找到 **A** 之中大於 **x** 的最小值。然而，這個函式有誤。請問下列哪個函式呼叫可測出函式有誤？

- (A) **Search(-1)**  
(B) **Search(0)**  
(C) **Search(10)**  
(D) **Search(16)**

```
int A[8]={0, 2, 4, 6, 8, 10, 12, 14};

int Search (int x) {
    int high = 7;
    int low = 0;
    while (high > low) {
        int mid = (high + low)/2;
        if (A[mid] <= x) {
            low = mid + 1;
        }
        else {
            high = mid;
        }
    }
    return A[high];
}
```

2. 給定函式 **A1()**、**A2()** 與 **F()** 如下，以下敘述何者有誤？

```
void A1 (int n) {
    F(n/5);
    F(4*n/5);
}
```

```
void A2 (int n) {
    F(2*n/5);
    F(3*n/5);
}
```

```
void F (int x) {
    int i;
    for (i=0; i<x; i=i+1)
        printf("*");
    if (x>1) {
        F(x/2);
        F(x/2);
    }
}
```

- (A) **A1(5)** 印的 '\*' 個數比 **A2(5)** 多  
(B) **A1(13)** 印的 '\*' 個數比 **A2(13)** 多  
(C) **A2(14)** 印的 '\*' 個數比 **A1(14)** 多  
(D) **A2(15)** 印的 '\*' 個數比 **A1(15)** 多

3. 右側 **F()** 函式回傳運算式該如何寫，才會使得 **F(14)** 的回傳值為 40?

- (A)  $n * F(n-1)$
- (B)  $n + F(n-3)$
- (C)  $n - F(n-2)$
- (D)  $F(3n+1)$

```
int F (int n) {  
    if (n < 4)  
        return n;  
    else  
        return _____?_____;  
}
```

4. 右側函式兩個回傳式分別該如何撰寫，才能正確計算並回傳兩參數 **a, b** 之最大公因數 (Greatest Common Divisor) ?

- (A) **a, GCD(b, r)**
- (B) **b, GCD(b, r)**
- (C) **a, GCD(a, r)**
- (D) **b, GCD(a, r)**

```
int GCD (int a, int b) {  
    int r;  
  
    r = a % b;  
    if (r == 0)  
        return _____;  
    return _____;  
}
```

5. 若 **A** 是一個可儲存 **n** 筆整數的陣列，且資料儲存於 **A[0]~A[n-1]**。經過右側程式碼運算後，以下何者敘述不一定正確？

- (A) **p** 是 **A** 陣列資料中的最大值
- (B) **q** 是 **A** 陣列資料中的最小值
- (C) **q < p**
- (D) **A[0] <= p**

```
int A[n]={ ... };  
int p = q = A[0];  
for (int i=1; i<n; i=i+1) {  
    if (A[i] > p)  
        p = A[i];  
    if (A[i] < q)  
        q = A[i];  
}
```

6. 若 **A[][]** 是一個  $M \times N$  的整數陣列，右側程式片段用以計算 A 陣列每一列的總和，以下敘述何者正確？

```
void main () {  
    int rowsum = 0;  
    for (int i=0; i<M; i=i+1) {  
        for (int j=0; j<N; j=j+1) {  
            rowsum = rowsum + A[i][j];  
        }  
        printf("The sum of row %d is %d.\n", i, rowsum);  
    }  
}
```

- (A) 第一列總和是正確，但其他列總和不一定正確  
(B) 程式片段在執行時會產生錯誤 (run-time error)  
(C) 程式片段中有語法上的錯誤  
(D) 程式片段會完成執行並正確印出每一列的總和

7. 若以 **B(5,2)** 呼叫右側 **B()** 函式，總共會印出幾次 “base case”？

```
int B (int n, int k) {  
    if (k == 0 || k == n){  
        printf ("base case\n");  
        return 1;  
    }  
    return B(n-1,k-1) + B(n-1,k);  
}
```

- (A) 1  
(B) 5  
(C) 10  
(D) 19

8. 給定右側程式，其中 `s` 有被宣告為全域變數，請問程式執行後輸出為何？

- (A) 1, 6, 7, 7, 8, 8, 9
- (B) 1, 6, 7, 7, 8, 1, 9
- (C) 1, 6, 7, 8, 9, 9, 9
- (D) 1, 6, 7, 7, 8, 9, 9

```
int s = 1; // 全域變數

void add (int a) {
    int s = 6;
    for( ; a>=0; a=a-1) {
        printf("%d,", s);
        s++;
        printf("%d,", s);
    }
}

int main () {
    printf("%d,", s);
    add(s);
    printf("%d,", s);
    s = 9;
    printf("%d", s);
    return 0;
}
```

9. 右側 `F()` 函式執行時，若輸入依序為整數 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，請問 `X[]` 陣列的元素值依順序為何？

- (A) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- (B) 2, 0, 2, 0, 2, 0, 2, 0, 2, 0
- (C) 9, 0, 1, 2, 3, 4, 5, 6, 7, 8
- (D) 8, 9, 0, 1, 2, 3, 4, 5, 6, 7

```
void F () {
    int X[10] = {0};
    for (int i=0; i<10; i=i+1) {
        scanf("%d", &X[(i+2)%10]);
    }
}
```

10. 若以 `G(100)` 呼叫右側函式後，`n` 的值為何？

- (A) 25
- (B) 75
- (C) 150
- (D) 250

```
int n = 0;

void K (int b) {
    n = n + 1;
    if (b % 4)
        K(b+1);
}

void G (int m) {
    for (int i=0; i<m; i=i+1) {
        K(i);
    }
}
```



11. 若 **A[1]**、**A[2]**，和 **A[3]** 分別為陣列 **A[]** 的三個元素(element)，下列那個程式片段可以將 **A[1]** 和 **A[2]** 的內容交換？

- (A) **A[1] = A[2]; A[2] = A[1];**
- (B) **A[3] = A[1]; A[1] = A[2]; A[2] = A[3];**
- (C) **A[2] = A[1]; A[3] = A[2]; A[1] = A[3];**
- (D) 以上皆可

12. 若函式 **rand()** 的回傳值為一介於 0 和 10000 之間的亂數，下列那個運算式可產生介於 100 和 1000 之間的任意數(包含 100 和 1000)？

- (A) **rand() % 900 + 100**
- (B) **rand() % 1000 + 1**
- (C) **rand() % 899 + 101**
- (D) **rand() % 901 + 100**

13. 右側程式片段無法正確列印 20 次的"Hi!"，請問下列哪一個修正方式仍無法正確列印 20 次的"Hi!"？

```
for (int i=0; i<=100; i=i+5) {  
    printf ("%s\n", "Hi!");  
}
```

- (A) 需要將 **i<=100** 和 **i=i+5** 分別修正為 **i<20** 和 **i=i+1**
- (B) 需要將 **i=0** 修正為 **i=5**
- (C) 需要將 **i<=100** 修正為 **i<100**;
- (D) 需要將 **i=0** 和 **i<=100** 分別修正為 **i=5** 和 **i<100**

14. 若以 **F(15)** 呼叫右側 **F()** 函式，總共會印出幾行數字？

- (A) 16 行
- (B) 22 行
- (C) 11 行
- (D) **15 行**

```
void F (int n) {  
    printf ("%d\n" , n);  
    if ((n%2 == 1) && (n > 1)){  
        return F(5*n+1);  
    }  
    else {  
        if (n%2 == 0)  
            return F(n/2);  
    }  
}
```

15. 給定右側函式  $F()$ ，執行  $F()$  時哪一行程式碼可能永遠不會被執行到？

- (A)  $a = a + 5;$
- (B)  $a = a + 2;$
- (C)  $a = 5;$
- (D) 每一行都執行得到

```
void F (int a) {  
    while (a < 10)  
        a = a + 5;  
    if (a < 12)  
        a = a + 2;  
    if (a <= 11)  
        a = 5;  
}
```

16. 給定右側函式  $F()$ ，已知  $F(7)$  回傳值為 17，且  $F(8)$  回傳值為 25，請問 **if** 的條件判斷式應為何？

- (A)  $a \% 2 \neq 1$
- (B)  $a * 2 > 16$
- (C)  $a + 3 < 12$
- (D)  $a * a < 50$

```
int F (int a) {  
    if ( _____?_____ )  
        return a * 2 + 3;  
    else  
        return a * 3 + 1;  
}
```

17. 給定右側函式  $F()$ ， $F()$  執行完所回傳的  $x$  值為何？

- (A)  $n(n+1)\sqrt{\lceil \log_2 n \rceil}$
- (B)  $n^2(n+1)/2$
- (C)  $n(n+1)\lceil \log_2 n + 1 \rceil / 2$
- (D)  $n(n+1)/2$

```
int F (int n) {  
    int x = 0;  
    for (int i=1; i<=n; i=i+1)  
        for (int j=i; j<=n; j=j+1)  
            for (int k=1; k<=n; k=k*2)  
                x = x + 1;  
    return x;  
}
```

18. 右側程式執行完畢後所輸出值為何？

- (A) 12
- (B) 24
- (C) 16
- (D) 20

```
int main() {  
    int x = 0, n = 5;  
    for (int i=1; i<=n; i=i+1)  
        for (int j=1; j<=n; j=j+1) {  
            if ((i+j)==2)  
                x = x + 2;  
            if ((i+j)==3)  
                x = x + 3;  
            if ((i+j)==4)  
                x = x + 4;  
        }  
    printf ("%d\n", x);  
    return 0;  
}
```

19. 右側程式擬找出陣列 **A[]** 中的最大值和最小值。不過，這段程式碼有誤，請問 **A[]** 初始值如何設定就可以測出程式有誤？

- (A) {90, 80, 100}
- (B) {80, 90, 100}
- (C) {100, 90, 80}
- (D) {90, 100, 80}

```
int main () {  
    int M = -1, N = 101, s = 3;  
    int A[] = _____?_____;  
  
    for (int i=0; i<s; i=i+1) {  
        if (A[i]>M) {  
            M = A[i];  
        }  
        else if (A[i]<N) {  
            N = A[i];  
        }  
    }  
    printf("M = %d, N = %d\n", M, N);  
    return 0;  
}
```

20. 小藍寫了一段複雜的程式碼想考考你是否了解函式的執行流程。請回答程式最後輸出的數值為何？

- (A) 70
- (B) 80
- (C) 100
- (D) 190

```
int g1 = 30, g2 = 20;

int f1(int v) {
    int g1 = 10;
    return g1+v;
}

int f2(int v) {
    int c = g2;
    v = v+c+g1;
    g1 = 10;
    c = 40;
    return v;
}

int main() {
    g2 = 0;
    g2 = f1(g2);
    printf("%d", f2(f2(g2)));
    return 0;
}
```

21. 若以 **F(5,2)** 呼叫右側 **F()** 函式，執行完畢後回傳值為何？

- (A) 1
- (B) 3
- (C) 5
- (D) 8

```
int F (int x,int y) {
    if (x<1)
        return 1;
    else
        return F(x-y,y)+F(x-2*y,y);
}
```

22. 若要邏輯判斷式 **!(X<sub>1</sub> || X<sub>2</sub>)** 計算結果為真(True)，則 **X<sub>1</sub>** 與 **X<sub>2</sub>** 的值分別應為何？

- (A) **X<sub>1</sub>** 為 False，**X<sub>2</sub>** 為 False
- (B) **X<sub>1</sub>** 為 True，**X<sub>2</sub>** 為 True
- (C) **X<sub>1</sub>** 為 True，**X<sub>2</sub>** 為 False
- (D) **X<sub>1</sub>** 為 False，**X<sub>2</sub>** 為 True

23. 程式執行時，程式中的變數值是存放在

- (A) 記憶體
- (B) 硬碟
- (C) 輸出入裝置
- (D) 匯流排

24. 程式執行過程中，若變數發生溢位情形，其主要原因為何？

- (A) 以有限數目的位元儲存變數值
- (B) 電壓不穩定
- (C) 作業系統與程式不甚相容
- (D) 變數過多導致編譯器無法完全處理

25. 若  $a, b, c, d, e$  均為整數變數，下列哪個算式計算結果與  $a+b*c-e$  計算結果相同？

- (A)  $((a+b)*c)-e$
- (B)  $((a+b)*(c-e))$
- (C)  $((a+(b*c))-e)$
- (D)  $(a+((b*c)-e))$

## 第 1 題 秘密差

### 問題描述

將一個十進位正整數的奇數位數的和稱為  $A$ ，偶數位數的和稱為  $B$ ，則  $A$  與  $B$  的絕對差值  $|A-B|$  稱為這個正整數的秘密差。

例如：263541 的奇數位數的和  $A = 6+5+1 = 12$ ，偶數位數的和  $B = 2+3+4 = 9$ ，所以 263541 的秘密差是  $|12-9| = 3$ 。

給定一個十進位正整數  $X$ ，請找出  $X$  的秘密差。

### 輸入格式

輸入為一行含有一個十進位表示法的正整數  $X$ ，之後是一個換行字元。

### 輸出格式

請輸出  $X$  的秘密差  $Y$  (以十進位表示法輸出)，以換行字元結尾。

### 範例一：輸入

263541

### 範例一：正確輸出

3

(說明) 263541 的  $A = 6+5+1 = 12$ ， $B = 2+3+4 = 9$ ， $|A-B| = |12-9| = 3$ 。

### 範例二：輸入

131

### 範例二：正確輸出

1

(說明) 131 的  $A = 1+1 = 2$ ， $B = 3$ ， $|A-B| = |2-3| = 1$ 。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $X$  一定恰好四位數。

第 2 子題組 30 分： $X$  的位數不超過 9。

第 3 子題組 50 分： $X$  的位數不超過 1000。

## 第 2 題 小群體

### 問題描述

Q 同學正在學習程式，P 老師出了以下的題目讓他練習。

一群人在一起時經常會形成一個一個的小群體。假設有  $N$  個人，編號由 0 到  $N-1$ ，每個人都寫下他最好朋友的編號（好朋友有可能是他自己的編號，如果他自己沒有其他好友），在本題中，**每個人的好友編號絕對不會重複，也就是說 0 到  $N-1$  每個數字都恰好出現一次。**

這種好友的關係會形成一些小群體。例如  $N=10$ ，好友編號如下，

	0	1	2	3	4	5	6	7	8	9
好友編號	4	7	2	9	6	0	8	1	5	3

0 的好友是 4，4 的好友是 6，6 的好友是 8，8 的好友是 5，5 的好友是 0，所以 0、4、6、8、和 5 就形成了一個小群體。另外，1 的好友是 7 而且 7 的好友是 1，所以 1 和 7 形成另一個小群體，同理，3 和 9 是一個小群體，而 2 的好友是自己，因此他自己是一個小群體。總而言之，在這個例子裡有 4 個小群體： $\{0,4,6,8,5\}$ 、 $\{1,7\}$ 、 $\{3,9\}$ 、 $\{2\}$ 。本題的問題是：輸入每個人的好友編號，計算出總共有幾個小群體。

Q 同學想了想卻不知如何下手，和藹可親的 P 老師於是給了他以下的提示：如果你從任何一人  $x$  開始，追蹤他的好友，好友的好友，....，這樣一直下去，一定會形成一個圈回到  $x$ ，這就是一個小群體。如果我們追蹤的過程中把追蹤過的加以標記，很容易知道哪些人已經追蹤過，因此，當一個小群體找到之後，我們再從任何一個還未追蹤過的開始繼續找下一個小群體，直到所有的人都追蹤完畢。

Q 同學聽完之後很順利的完成了作業。

在本題中，你的任務與 Q 同學一樣：給定一群人的好友，請計算出小群體個數。

### 輸入格式

第一行是一個正整數  $N$ ，說明團體中人數。

第二行依序是 0 的好友編號、1 的好友編號、.....、 $N-1$  的好友編號。共有  $N$  個數字，包含 0 到  $N-1$  的每個數字恰好出現一次，數字間會有一個空白隔開。

### 輸出格式

請輸出小群體的個數。不要有任何多餘的字或空白，並以換行字元結尾。

#### 範例一：輸入

10  
4 7 2 9 6 0 8 1 5 3

#### 範例一：正確輸出

4

#### （說明）

4 個小群體是  $\{0,4,6,8,5\}$ ， $\{1,7\}$ ， $\{3,9\}$  和  $\{2\}$ 。

#### 範例二：輸入

3  
0 2 1

#### 範例二：正確輸出

2

#### （說明）

2 個小群體分別是  $\{0\}$ ， $\{1,2\}$ 。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $1 \leq N \leq 100$ ，每一個小群體不超過 2 人。

第 2 子題組 30 分， $1 \leq N \leq 1,000$ ，無其他限制。

第 3 子題組 50 分， $1,001 \leq N \leq 50,000$ ，無其他限制。



## 第 3 題 數字龍捲風

### 問題描述

給定一個  $N \times N$  的二維陣列，其中  $N$  是奇數，我們可以從正中間的位置開始，以順時針旋轉的方式走訪每個陣列元素恰好一次。對於給定的陣列內容與起始方向，請輸出走訪順序之內容。下面的例子顯示了  $N=5$  且第一步往左的走訪順序：

3	4	2	1	4
4	2	3	8	9
2	1	9	5	6
4	2	3	7	8
1	2	6	4	3

依此順序輸出陣列內容則可以得到「9123857324243421496834621」。

類似地，如果是第一步向上，則走訪順序如下：

3	4	2	1	4
4	2	3	8	9
2	1	9	5	6
4	2	3	7	8
1	2	6	4	3

依此順序輸出陣列內容則可以得到「9385732124214968346214243」。

### 輸入格式

輸入第一行是整數  $N$ ， $N$  為奇數且不小於 3。第二行是一個 0~3 的整數代表起始方向，其中 0 代表左、1 代表上、2 代表右、3 代表下。第三行開始  $N$  行是陣列內容，順序是由上而下，由左至右，陣列的內容為 0~9 的整數，同一行數字中間以一個空白間隔。

### 輸出格式

請輸出走訪順序的陣列內容，該答案會是一連串的數字，數字之間不要輸出空白，結尾有換行符號。

#### 範例一：輸入

```
5
0
3 4 2 1 4
4 2 3 8 9
2 1 9 5 6
4 2 3 7 8
1 2 6 4 3
```

#### 範例一：正確輸出

```
9123857324243421496834621
```

#### 範例二：輸入

```
3
1
4 1 2
3 0 5
6 7 8
```

#### 範例二：正確輸出

```
012587634
```

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $3 \leq N \leq 5$ ，且起始方向均為向左。

第 2 子題組 80 分， $3 \leq N \leq 49$ ，起始方向無限定。

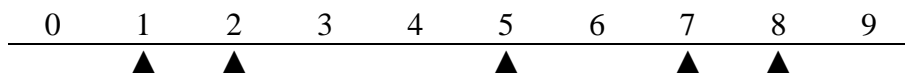
提示：本題有多種處理方式，其中之一是觀察每次轉向與走的步數。例如，起始方向是向左時，前幾步的走法是：左 1、上 1、右 2、下 2、左 3、上 3、.....一直到出界為止。

## 第 4 題 基地台

### 問題描述

為因應資訊化與數位化的發展趨勢，某市長想要在城市的一些服務點上提供無線網路服務，因此他委託電信公司架設無線基地台。某電信公司負責其中  $N$  個服務點，這  $N$  個服務點位在一條筆直的大道上，它們的位置(座標)係以與該大道一端的距離  $P[i]$  來表示，其中  $i=0\sim N-1$ 。由於設備訂製與維護的因素，每個基地台的服務範圍必須都一樣，當基地台架設後，與此基地台距離不超過  $R$  (稱為基地台的半徑)的服務點都可以使用無線網路服務，也就是說每一個基地台可以服務的範圍是  $D=2R$  (稱為基地台的直徑)。現在電信公司想要計算，如果要架設  $K$  個基地台，那麼基地台的最小直徑是多少才能使每個服務點都可以得到服務。

基地台架設的地點不一定要在服務點上，最佳的架設地點也不唯一，但本題只要求最小直徑即可。以下是一個  $N=5$  的例子，五個服務點的座標分別是 1、2、5、7、8。



假設  $K=1$ ，最小的直徑是 7，基地台架設在座標 4.5 的位置，所有點與基地台的距離都在半徑 3.5 以內。假設  $K=2$ ，最小的直徑是 3，一個基地台服務座標 1 與 2 的點，另一個基地台服務另外三點。在  $K=3$  時，直徑只要 1 就足夠了。

### 輸入格式

輸入有兩行。第一行是兩個正整數  $N$  與  $K$ ，以一個空白間格。第二行  $N$  個非負整數  $P[0], P[1], \dots, P[N-1]$  表示  $N$  個服務點的位置，這些位置彼此之間以一個空白間格。請注意，這  $N$  個位置並不保證相異也未經過排序。本題中， $K < N$  且所有座標是整數，因此，所求最小直徑必然是不小於 1 的整數。

### 輸出格式

輸出最小直徑，不要有任何多餘的字或空白並以換行結尾。

#### 範例一：輸入

5 2  
5 1 2 8 7

#### 範例一：正確輸出

3  
(說明) 如題目中之說明。

#### 範例二：輸入

5 1  
7 5 1 2 8

#### 範例二：正確輸出

7  
(說明) 如題目中之說明。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 2 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分，座標範圍不超過 100， $1 \leq K \leq 2$ ， $K < N \leq 10$ 。

第 2 子題組 20 分，座標範圍不超過 1,000， $1 \leq K < N \leq 100$ 。

第 3 子題組 20 分，座標範圍不超過 1,000,000,000， $1 \leq K < N \leq 500$ 。

第 4 子題組 50 分，座標範圍不超過 1,000,000,000， $1 \leq K < N \leq 50,000$ 。

## 第 1 題 三角形辨別

### 問題描述

三角形除了是最基本的多邊形外，亦可進一步細分為鈍角三角形、直角三角形及銳角三角形。若給定三個線段的長度，透過下列公式的運算，即可得知此三線段能否構成三角形，亦可判斷是直角、銳角和鈍角三角形。

提示：若  $a$ 、 $b$ 、 $c$  為三個線段的邊長，且  $c$  為最大值，則

- 若  $a + b \leq c$ ，三線段無法構成三角形
- 若  $a \times a + b \times b < c \times c$ ，三線段構成鈍角三角形 (Obtuse triangle)
- 若  $a \times a + b \times b = c \times c$ ，三線段構成直角三角形 (Right triangle)
- 若  $a \times a + b \times b > c \times c$ ，三線段構成銳角三角形 (Acute triangle)

請設計程式以讀入三個線段的長度判斷並輸出此三線段可否構成三角形？若可，判斷並輸出其所屬三角形類型。

### 輸入格式

輸入僅一行包含三正整數，三正整數皆小於 30,001，兩數之間有一空白。

### 輸出格式

輸出共有兩行，第一行由小而大印出此三正整數，兩數字之間以一個空白間格，最後一個數字後不應有空白；第二行輸出三角形的類型：

- 若無法構成三角形時輸出「No」；
- 若構成鈍角三角形時輸出「Obtuse」；
- 若直角三角形時輸出「Right」；
- 若銳角三角形時輸出「Acute」。

#### 範例一：輸入

3 4 5

#### 範例一：正確輸出

3 4 5

Right

(說明)  $a \times a + b \times b = c \times c$   
成立時為直角三角形。

#### 範例二：輸入

101 100 99

#### 範例二：正確輸出

99 100 101

Acute

(說明) 邊長排序由小到大輸出， $a \times a + b \times b > c \times c$   
成立時為銳角三角形。

#### 範例三：輸入

10 100 10

#### 範例三：正確輸出

10 10 100

No

(說明) 由於無法構成三角形，因此第二行須印出「No」。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。

## 第 2 題 最大和

### 問題描述

給定  $N$  群數字，每群都恰有  $M$  個正整數。若從每群數字中各選擇一個數字（假設第  $i$  群所選出數字為  $t_i$ ），將所選出的  $N$  個數字加總即可得總和  $S = t_1 + t_2 + \dots + t_N$ 。請寫程式計算  $S$  的最大值（最大總和），並判斷各群所選出的數字是否可以整除  $S$ 。

### 輸入格式

第一行有二個正整數  $N$  和  $M$ ， $1 \leq N \leq 20$ ， $1 \leq M \leq 20$ 。

接下來的  $N$  行，每一行各有  $M$  個正整數  $x_i$ ，代表一群整數，數字與數字間有一個空格，且  $1 \leq i \leq M$ ，以及  $1 \leq x_i \leq 256$ 。

### 輸出格式

第一行輸出最大總和  $S$ 。

第二行按照被選擇數字所屬群的順序，輸出可以整除  $S$  的被選擇數字，數字與數字間以一個空格隔開，最後一個數字後無空白；若  $N$  個被選擇數字都不能整除  $S$ ，就輸出 -1。

#### 範例一：輸入

```
3 2
1 5
6 4
1 1
```

#### 範例一：正確輸出

```
12
6 1
```

（說明）挑選的數字依序是 5, 6, 1，總和  $S=12$ 。而此三數中可整除  $S$  的是 6 與 1，6 在第二群，1 在第 3 群所以先輸出 6 再輸出 1。注意，1 雖然也出現在第一群，但她不是第一群中挑出的數字，所以順序是先 6 後 1。

#### 範例二：輸入

```
4 3
6 3 2
2 7 9
4 7 1
9 5 3
```

#### 範例二：正確輸出

```
31
-1
```

（說明）挑選的數字依序是 6, 9, 7, 9，總和  $S=31$ 。而此四數中沒有可整除  $S$  的，所以第二行輸出 -1。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分： $1 \leq N \leq 20$ ， $M = 1$ 。

第 2 子題組 30 分： $1 \leq N \leq 20$ ， $M = 2$ 。

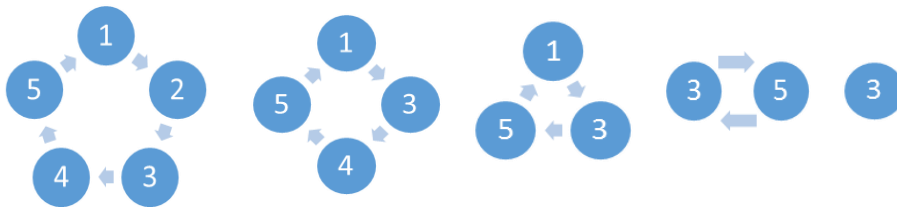
第 3 子題組 50 分： $1 \leq N \leq 20$ ， $1 \leq M \leq 20$ 。

## 第 3 題 定時 K 彈

### 問題描述

「定時 K 彈」是一個團康遊戲，N 個人圍成一個圈，由 1 號依序到 N 號，從 1 號開始依序傳遞一枚玩具炸彈，炸彈每次到第 M 個人就會爆炸，此人即淘汰，被淘汰的人要離開圓圈，然後炸彈再從該淘汰者的下一個開始傳遞。遊戲之所以稱 K 彈是因為這枚炸彈只會爆炸 K 次，在第 K 次爆炸後，遊戲即停止，而此時在第 K 個淘汰者的下一位遊戲者被稱為幸運者，通常就會被要求表演節目。例如  $N=5$ ， $M=2$ ，如果  $K=2$ ，炸彈會爆炸兩次，被爆炸淘汰的順序依序是 2 與 4（參見下圖），這時 5 號就是幸運者。如果  $K=3$ ，剛才的遊戲會繼續，第三個淘汰的是 1 號，所以幸運者是 3 號。如果  $K=4$ ，下一輪淘汰 5 號，所以 3 號是幸運者。

給定 N、M 與 K，請寫程式計算出誰是幸運者。



### 輸入格式

輸入只有一行包含三個正整數，依序為 N、M 與 K，兩數中間有一個空格分開。其中  $1 \leq K < N$ 。

### 輸出格式

請輸出幸運者的號碼，結尾有換行符號。

#### 範例一：輸入

5 2 4

#### 範例一：正確輸出

3

#### （說明）

被淘汰的順序是 2、4、1、5，此時 5 的下一位是 3，也是最後剩下的，所以幸運者是 3。

#### 範例二：輸入

8 3 6

#### 範例二：正確輸出

4

#### （說明）

被淘汰的順序是 3、6、1、5、2、8，此時 8 的下一位是 4，所以幸運者是 4。

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $1 \leq N \leq 100$ ，且  $1 \leq M \leq 10$ ， $K = N-1$ 。

第 2 子題組 30 分， $1 \leq N \leq 10,000$ ，且  $1 \leq M \leq 1,000,000$ ， $K = N-1$ 。

第 3 子題組 20 分， $1 \leq N \leq 200,000$ ，且  $1 \leq M \leq 1,000,000$ ， $K = N-1$ 。

第 4 子題組 30 分， $1 \leq N \leq 200,000$ ，且  $1 \leq M \leq 1,000,000$ ， $1 \leq K < N$ 。

## 第 4 題 棒球遊戲

### 問題描述

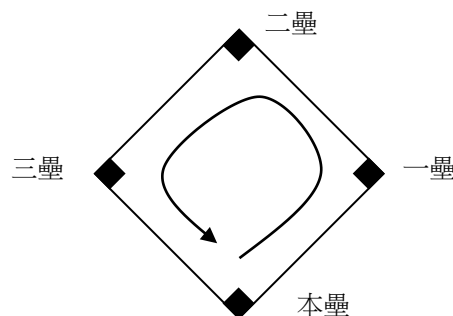
謙謙最近迷上棒球，他想自己寫一個簡化的棒球遊戲計分程式。這個程式會讀入球隊中每位球員的打擊結果，然後計算出球隊的得分。

這是個簡化版的模擬，假設擊球員的打擊結果只有以下情況：

- (1) 安打：以 1B, 2B, 3B 和 HR 分別代表一壘打、二壘打、三壘打和全（四）壘打。
- (2) 出局：以 FO, GO, 和 SO 表示。

這個簡化版的規則如下：

- (1) 球場上有四個壘包，稱為本壘、一壘、二壘和三壘。
- (2) 站在本壘握著球棒打球的稱為「擊球員」，站在另外三個壘包的稱為「跑壘員」。
- (3) 當擊球員的打擊結果為「安打」時，場上球員（擊球員與跑壘員）可以移動；結果為「出局」時，跑壘員不動，擊球員離場，換下一位擊球員。
- (4) 球隊總共有九位球員，依序排列。比賽開始由第 1 位開始打擊，當第  $i$  位球員打擊完畢後，由第  $(i+1)$  位球員擔任擊球員。當第九位球員完畢後，則輪回第一位球員。
- (5) 當打出  $K$  壘打時，場上球員（擊球員和跑壘員）會前進  $K$  個壘包。從本壘前進一個壘包會移動到一壘，接著是二壘、三壘，最後回到本壘。
- (6) 每位球員回到本壘時可得 1 分。
- (7) 每達到三個出局數時，一、二和三壘就會清空（跑壘員都得離開），重新開始。



請寫出具備這樣功能的程式，計算球隊的總得分。

### 輸入格式

1. 每組測試資料固定有十行。
2. 第一到九行，依照球員順序，每一行代表一位球員的打擊資訊。每一行開始有一個正整數  $a$  ( $1 \leq a \leq 5$ )，代表球員總共打了  $a$  次。接下來有  $a$  個字串（均為兩個字元），依序代表每次打擊的結果。資料之間均以一個空白字元隔開。球員的打擊資訊不會有錯誤也不會缺漏。
3. 第十行有一個正整數  $b$  ( $1 \leq b \leq 27$ )，表示我們想要計算當總出局數累計到  $b$  時，該球隊的得分。輸入的打擊資訊中至少包含  $b$  個出局。



## 輸出格式

計算在總計第  $b$  個出局數發生時的總得分，並將此得分輸出於一行。

### 範例一：輸入

```
5 1B 1B FO GO 1B
5 1B 2B FO FO SO
4 SO HR SO 1B
4 FO FO FO HR
4 1B 1B 1B 1B
4 GO GO 3B GO
4 1B GO GO SO
4 SO GO 2B 2B
4 3B GO GO FO
3
```

### 範例一：正確輸出

0

(說明)

1B：一壘有跑壘員。

1B：一、二壘有跑壘員。

SO：一、二壘有跑壘員，一出局。

FO：一、二壘有跑壘員，兩出局。

1B：一、二、三壘有跑壘員，兩出局。

GO：一、二、三壘有跑壘員，三出局。

達到第三個出局數時，一、二、三壘均有跑壘員，但無法得分。因為  $b=3$ ，代表三個出局就結束比賽，因此得到 0 分。

### 範例二：輸入

```
5 1B 1B FO GO 1B
5 1B 2B FO FO SO
4 SO HR SO 1B
4 FO FO FO HR
4 1B 1B 1B 1B
4 GO GO 3B GO
4 1B GO GO SO
4 SO GO 2B 2B
4 3B GO GO FO
6
```

### 範例二：正確輸出

5

(說明) 接續範例一，達到第三個出局數時未得分，壘上清空。

1B：一壘有跑壘員。

SO：一壘有跑壘員，一出局。

3B：三壘有跑壘員，一出局，得一分。

1B：一壘有跑壘員，一出局，得兩分。

2B：二、三壘有跑壘員，一出局，得兩分。

HR：一出局，得五分。

FO：兩出局，得五分。

1B：一壘有跑壘員，兩出局，得五分。

GO：一壘有跑壘員，三出局，得五分。

因為  $b=6$ ，代表要計算的是累積六個出局時的得分，因此在前 3 個出局數時得 0 分，第 4~6 個出局數得到 5 分，因此總得分是  $0+5=5$  分。

## 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分，打擊表現只有 HR 和 SO 兩種。

第 2 子題組 20 分，安打表現只有 1B，而且  $b$  固定為 3。

第 3 子題組 20 分， $b$  固定為 3。

第 4 子題組 40 分，無特別限制。



# 大學程式設計先修檢測

2017.10.28

## 實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

# 邏輯運算子 (Logic Operators)

## 問題描述

小蘇最近在學三種邏輯運算子 AND、OR 和 XOR。這三種運算子都是二元運算子，也就是說在運算時需要兩個運算元，例如  $a \text{ AND } b$ 。對於整數  $a$  與  $b$ ，以下三個二元運算子的運算結果定義如下列三個表格：

a AND b		
	b 為 0	b 不為 0
a 為 0	0	0
a 不為 0	0	1

a OR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	1

a XOR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	0

舉例來說：

- (1)  $0 \text{ AND } 0$  的結果為 0， $0 \text{ OR } 0$  以及  $0 \text{ XOR } 0$  的結果也為 0。
- (2)  $0 \text{ AND } 3$  的結果為 0， $0 \text{ OR } 3$  以及  $0 \text{ XOR } 3$  的結果則為 1。
- (3)  $4 \text{ AND } 9$  的結果為 1， $4 \text{ OR } 9$  的結果也為 1，但  $4 \text{ XOR } 9$  的結果為 0。

請撰寫一個程式，讀入  $a$ 、 $b$  以及邏輯運算的結果，輸出可能的邏輯運算為何。

## 輸入格式

輸入只有一行，共三個整數值，整數間以一個空白隔開。第一個整數代表  $a$ ，第二個整數代表  $b$ ，這兩數均為非負的整數。第三個整數代表邏輯運算的結果，只會是 0 或 1。

## 輸出格式

輸出可能得到指定結果的運算，若有多個，輸出順序為 AND、OR、XOR，每個可能的運算單獨輸出一行，每行結尾皆有換行。若不可能得到指定結果，輸出 IMPOSSIBLE。（注意輸出時所有英文字母均為大寫字母。）

### 範例一：輸入

0 0 0

### 範例一：正確輸出

AND  
OR  
XOR

### 範例二：輸入

1 1 1

### 範例二：正確輸出

AND  
OR

### 範例三：輸入

3 0 1

### 範例三：正確輸出

OR  
XOR

### 範例四：輸入

0 0 1

### 範例四：正確輸出

IMPOSSIBLE

## 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 80 分， $a$  和  $b$  的值只會是 0 或 1。

第 2 子題組 20 分， $0 \leq a, b < 10,000$ 。

# 大學程式設計先修檢測

2017.10.28

## 實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

# 交錯字串 (Alternating Strings)

## 問題描述

一個字串如果全由大寫英文字母組成，我們稱為大寫字串；如果全由小寫字母組成則稱為小寫字串。字串的長度是它所包含字母的個數，在本題中，字串均由大小寫英文字母組成。假設  $k$  是一個自然數，一個字串被稱為「 $k$ -交錯字串」，如果它是由長度為  $k$  的大寫字串與長度為  $k$  的小寫字串交錯串接組成。

舉例來說，「StRiNg」是一個 1-交錯字串，因為它是一個大寫一個小寫交替出現；而「heLLow」是一個 2-交錯字串，因為它是兩個小寫接兩個大寫再接兩個小寫。但不管  $k$  是多少，「aBBaaa」、「BaBaBB」、「aaaAAbbCCCC」都不是  $k$ -交錯字串。

本題的目標是對於給定  $k$  值，在一個輸入字串找出最長一段連續子字串滿足  $k$ -交錯字串的要求。例如  $k=2$  且輸入「aBBaaa」，最長的  $k$ -交錯字串是「BBaa」，長度為 4。又如  $k=1$  且輸入「BaBaBB」，最長的  $k$ -交錯字串是「BaBaB」，長度為 5。

請注意，滿足條件的子字串可能只包含一段小寫或大寫字母而無交替，如範例二。此外，也可能不存在滿足條件的子字串，如範例四。

## 輸入格式

輸入的第一行是  $k$ ，第二行是輸入字串，字串長度至少為 1，只由大小寫英文字母組成(A~Z, a~z)並且沒有空白。

## 輸出格式

輸出輸入字串中滿足  $k$ -交錯字串的要求的最長一段連續子字串的長度，以換行結尾。

### 範例一：輸入

1  
aBBdaaa

### 範例一：正確輸出

2

### 範例三：輸入

2  
aaFAXbbCDCCC

### 範例三：正確輸出

8

### 範例二：輸入

3  
DDaasAAbbCC

### 範例二：正確輸出

3

### 範例四：輸入

3  
DDaaAAbbCC

### 範例四：正確輸出

0

## 評分說明：

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分，字串長度不超過 20 且  $k=1$ 。

第 2 子題組 30 分，字串長度不超過 100 且  $k \leq 2$ 。

第 3 子題組 50 分，字串長度不超過 100,000 且無其他限制。

**提示：**根據定義，要找的答案是大寫片段與小寫片段交錯串接而成。本題有多種解法的思考方式，其中一種是從左往右掃描輸入字串，我們需要紀錄的狀態包含：目前是在小寫子字串中還是大寫子字串中，以及在目前大(小)寫子字串的第幾個位置。根據下一個字母的

大小寫，我們需要更新狀態並且記錄以此位置為結尾的最長交替字串長度。

另外一種思考是先掃描一遍字串，找出每一個連續大(小)寫片段的長度並將其記錄在一個陣列，然後針對這個陣列來找出答案。

# 大學程式設計先修檢測

2017.10.28

## 實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

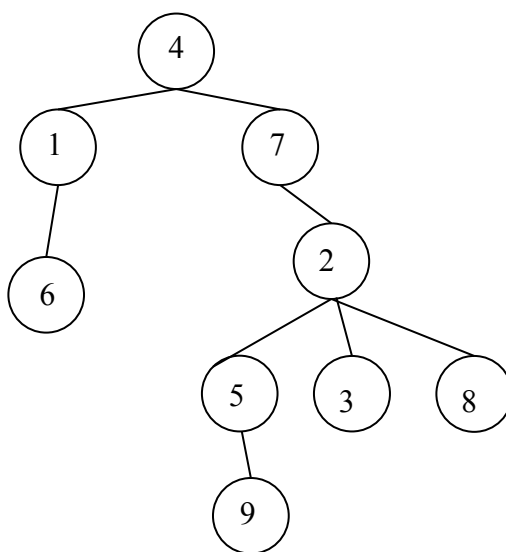
2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

## 樹狀圖分析 (Tree Analyses)

### 問題描述

本題是關於有根樹(rooted tree)。在一棵  $n$  個節點的有根樹中，每個節點都是以  $1 \sim n$  的不同數字來編號，描述一棵有根樹必須定義節點與節點之間的親子關係。一棵有根樹恰有一個節點沒有父節點(parent)，此節點被稱為根節點(root)，除了根節點以外的每一個節點都恰有一個父節點，而每個節點被稱為是它父節點的子節點(child)，有些節點沒有子節點，這些節點稱為葉節點(leaf)。在當有根樹只有一個節點時，這個節點既是根節點同時也是葉節點。

在圖形表示上，我們將父節點畫在子節點之上，中間畫一條邊(edge)連結。例如，圖一中表示的是一棵 9 個節點的有根樹，其中，節點 1 為節點 6 的父節點，而節點 6 為節點 1 的子節點；又 5、3 與 8 都是 2 的子節點。節點 4 沒有父節點，所以節點 4 是根節點；而 6、9、3 與 8 都是葉節點。



圖一

樹狀圖中的兩個節點  $u$  和  $v$  之間的距離  $d(u,v)$  定義為兩節點之間邊的數量。如圖一中， $d(7, 5) = 2$ ，而  $d(1, 2) = 3$ 。對於樹狀圖中的節點  $v$ ，我們以  $h(v)$  代表節點  $v$  的高度，其定義是節點  $v$  和節點  $v$  下面最遠的葉節點之間的距離，而葉節點的高度定義為 0。如圖一中，節點 6 的高度為 0，節點 2 的高度為 2，而節點 4 的高度為 4。此外，我們定義  $H(T)$  為  $T$  中所有節點的高度總和，也就是說  $H(T) = \sum_{v \in T} h(v)$ 。給定一個樹狀圖  $T$ ，請找出  $T$  的根節點以及高度總和  $H(T)$ 。

### 輸入格式

第一行有一個正整數  $n$  代表樹狀圖的節點個數，節點的編號為 1 到  $n$ 。接下來有  $n$  行，第  $i$  行的第一個數字  $k$  代表節點  $i$  有  $k$  個子節點，第  $i$  行接下來的  $k$  個數字就是這些子節點的編號。每一行的相鄰數字間以空白隔開。

### 輸出格式

輸出兩行各含一個整數，第一行是根節點的編號，第二行是  $H(T)$ 。

**範例一：輸入**

```
7
0
2 6 7
2 1 4
0
2 3 2
0
0
```

**範例一：正確輸出**

```
5
4
```

**範例二：輸入**

```
9
1 6
3 5 3 8
0
2 1 7
1 9
0
1 2
0
0
```

**範例二：正確輸出**

```
4
11
```

**評分說明：**

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。測資範圍如下，其中  $k$  是每個節點的子節點數量上限：

第 1 子題組 10 分， $1 \leq n \leq 4, k \leq 3$ ，除了根節點之外都是葉節點。

第 2 子題組 30 分， $1 \leq n \leq 1,000, k \leq 3$ 。

第 3 子題組 30 分， $1 \leq n \leq 100,000, k \leq 3$ 。

第 4 子題組 30 分， $1 \leq n \leq 100,000, k$  無限制。

**提示：**輸入的資料是給每個節點的子節點有哪些或沒有子節點，因此，可以根據定義找出根節點。關於節點高度的計算，我們根據定義可以找出以下遞迴關係式：(1)葉節點的高度為 0；(2)如果  $v$  不是葉節點，則  $v$  的高度是它所有子節點的最大高度加一。也就是說，假設  $v$  的子節點有  $a, b$  與  $c$ ，則  $h(v) = \max\{h(a), h(b), h(c)\} + 1$ 。以遞迴方式可以計算出所有節點的高度。



# 大學程式設計先修檢測

2017.10.28

## 實作題 注意事項

1. 請依題目敘述撰寫一完整程式，上傳送審之程式檔案副檔名必須為 .c, .cpp, .java, 或 .py，如下表所式。另若以 Java 撰寫程式，class 名稱必須與檔名 (p1, p2, p3, 或 p4) 一致。

	C	C++	Java	Python
邏輯運算子	p1.c	p1.cpp	p1.java	p1.py
交錯字串	p2.c	p2.cpp	p2.java	p2.py
樹狀圖分析	p3.c	p3.cpp	p3.java	p3.py
物品堆疊	p4.c	p4.cpp	p4.java	p4.py

2. 上傳程式檔案前，請自行測試程式是否能正常編譯，程式執行時輸入、輸出格式是否正確。評分時，若程式無法正常編譯或執行，將以 0 分計算。

# 物品堆疊(Stacking)

## 問題描述

某個自動化系統中有一個存取物品的子系統，該系統是將  $N$  個物品堆在一個垂直的貨架上，每個物品各佔一層。系統運作的方式如下：每次只會取用一個物品，取用時必須先將在其上方的物品貨架升高，取用後必須將該物品放回，然後將剛才升起的貨架降回原始位置，之後才會進行下一個物品的取用。

每一次升高某些物品所需要消耗的能量是以這些物品的總重來計算，在此我們忽略貨架的重量以及其他可能的消耗。現在有  $N$  個物品，第  $i$  個物品的重量是  $w(i)$  而需要取用的次數為  $f(i)$ ，我們需要決定如何擺放這些物品的順序來讓消耗的能量越小越好。舉例來說，有兩個物品  $w(1)=1$ 、 $w(2)=2$ 、 $f(1)=3$ 、 $f(2)=4$ ，也就是說物品 1 的重量是 1 需取用 3 次，物品 2 的重量是 2 需取用 4 次。我們有兩個可能的擺放順序(由上而下)：

- (1,2)，也就是物品 1 放在上方，2 在下方。那麼，取用 1 的時候不需要能量，而每次取用 2 的能量消耗是  $w(1)=1$ ，因為 2 需取用  $f(2)=4$  次，所以消耗能量數為  $w(1)*f(2)=4$ 。
- (2,1)，也就是物品 2 放在 1 的上方。那麼，取用 2 的時候不需要能量，而每次取用 1 的能量消耗是  $w(2)=2$ ，因為 1 需取用  $f(1)=3$  次，所以消耗能量數  $=w(2)*f(1)=6$ 。

在所有可能的兩種擺放順序中，最少的能量是 4，所以答案是 4。再舉一例，若有三物品而  $w(1)=3$ 、 $w(2)=4$ 、 $w(3)=5$ 、 $f(1)=1$ 、 $f(2)=2$ 、 $f(3)=3$ 。假設由上而下以 (3,2,1) 的順序，此時能量計算方式如下：取用物品 3 不需要能量，取用物品 2 消耗  $w(3)*f(2)=10$ ，取用物品 1 消耗  $(w(3)+w(2))*f(1)=9$ ，總計能量為 19。如果以 (1,2,3) 的順序，則消耗能量為  $3*2+(3+4)*3=27$ 。事實上，我們一共有  $3!=6$  種可能的擺放順序，其中順序 (3,2,1) 可以得到最小消耗能量 19。

## 輸入格式

輸入的第一行是物品件數  $N$ ，第二行有  $N$  個正整數，依序是各物品的重量  $w(1)$ 、 $w(2)$ 、...、 $w(N)$ ，重量皆不超過 1000 且以一個空白間隔。第三行有  $N$  個正整數，依序是各物品的取用次數  $f(1)$ 、 $f(2)$ 、...、 $f(N)$ ，次數皆為 1000 以內的正整數，以一個空白間隔。

## 輸出格式

輸出最小能量消耗值，以換行結尾。所求答案不會超過 63 個位元所能表示的正整數。

### 範例一(第 1、3 子題)：輸入

```
2
20 10
1 1
```

### 範例一：正確輸出

```
10
```

### 範例二(第 2、4 子題)：輸入

```
3
3 4 5
1 2 3
```

### 範例二：正確輸出

```
19
```

**評分說明：**輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 10 分， $N=2$ ，且取用次數  $f(1)=f(2)=1$ 。

第 2 子題組 20 分， $N=3$ 。

第 3 子題組 45 分， $N \leq 1,000$ ，且每一個物品  $i$  的取用次數  $f(i)=1$ 。

第 4 子題組 25 分， $N \leq 100,000$ 。