九十三學年度高級中學資訊學科能力競賽決賽

上機程式設計題

作答注意事項：

一、 對考題有任何疑義，請於考試開始後二個小時之內填寫「問題單」，交付監考人員轉送命題委員提出問題，逾時不予回覆。

二、 第一題到第五題每題 16 分，第六題 20 分，共 100 分。

三、 可選擇指定解題語言中任何一種語言解題。

四、 最後繳交編譯後之執行檔限定在 Windows XP 的命令提示字元下執行。(若使用 Turbo Pascal 或 Turbo C 來編譯的話，請使用「開始」→「執行」→「command」下來執行)

五、 各題執行檔檔名請設定如下：
考生編號_題號.exe
例如：101_1.exe

六、 各題原始碼檔名請設定如下：
考生編號_題號.解題語言附屬檔名
例如：101_1.c

七、 各題輸入資料檔名如下：
in_題號.txt
例如：in_1.txt

八、 各題輸入方式以讀檔方式為之，請以目前工作目錄（Current Working Directory）下的檔案名稱為讀取路徑。

九、 各題輸出方式為標準輸出（螢幕)。

十、 考試結束後，將不再允許更動及重新編譯程式。

十一、所有發展的程式必須在 30 秒以內於試場內的電腦輸出結果，否則不予計分。

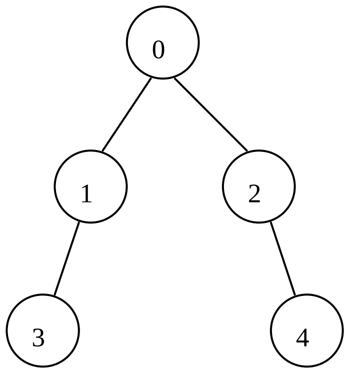
1. 銀河帝國旅行社

問題敘述：

很久很久以前，銀河系有一個科技極度發達的帝國存在，這個帝國有著許許多多的星球所組成，這些星球之間有主從關係，除了帝國首都之外，其他的星球均從屬於自己專屬的主星。

由於銀河是相當的浩瀚，主要的旅行方式是空間跳躍，不管這兩個星球有多遠只要花一小時就抵達目的地。爲了維護帝國間的和平與安定，帝國當局是嚴格的管制空間跳躍的使用，只允許主星到從屬星以及從屬星到主星之間的的空間跳躍。假定銀河帝國中，只有一個帝國首都，首都本身也是一個星球，每個星球只有一個專屬的主星，但一個星球可以有許多從屬星。並且不會發生循環從屬的關係，即不會發生如下情況：S_1是S_2的主星，S_2是S_3的主星...S_{n-1}是S_n的主星，S_n是S_1的主星。

以右圖爲例，1 號星與 2 號星爲 0 號星(也就是帝國首都)的從屬星，3 號星是 1 號星的從屬星，4 號星是 2 號星的從屬星。在這個例子中，從 3 號星到 2 號星作星際旅行需要 3 小時：3 號星 → 1 號星，1 號星 → 0 號星，0 號星 → 2 號星。而在這個星系中作星際旅行，最多需要花 4 小時，也就是從 3 號星旅行到 4 號星。



現在你的任務是估計銀河帝國之中，任意選兩個星球作爲起點與終點進行空間跳躍旅行，最多需要花多少時間？(以上圖爲例，答案是 4 個小時。)

輸入說明：

第一行將有一個數字 n (n≤10000)，代表銀河帝國中有 n 個星球。緊接下來會有 n 行，接下來的每行依序都代表了一個星球的從屬星名單。爲了簡化輸入的複雜度，將由 0 到 n−1 的數字來替代星球名稱，0 就是帝國首都，而且只有數字小的才可以當數字大的主星。每行的格式："從屬星 1　從屬星 2　…　-1"，最後的-1 代表從屬星名單結束，如果是單一一行的-1，代表該星球沒有從屬星。

輸出說明：

輸出一行，包含一個數字，即任選兩星球，最多需要花多少小時才能藉由空間跳

躍從起點抵達目的地。

輸入範例 1：(不包含//以後的的部分，該部分僅供註解用)：

```
5
1 2 -1      //首都(0 號星)的從屬星是 1 號星與 2 號星
3 -1        //1 號星的從屬星是 3 號星
4 -1        //2 號星的從屬星是 4 號星
-1          //3 號星沒有從屬星
-1          //4 號星沒有從屬星
```

輸出範例 1：

```
4
```

輸入範例 2：

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

輸出範例 2：

```
2
```

2. IC 板檢測

問題敘述：

「奇上」IC 板檢測工司擁有全國最多台的 IC 板自動檢測機，但是因爲每台檢測系統購進的時間不一樣，因此各機台檢測一片 IC 板所需時間也都不同。因爲該公司爲 24 小時營運，因此整天任何時刻都會有小貨車將需要檢測的 IC 板送至該公司進行檢測。每當一輛小貨車抵達時，「奇上」公司就立即安排一台 IC 板自動檢測系統來負責該批所有的 IC 板檢測工作。爲了爭取時效，「奇上」公司一定會安排能最快完成該批 IC 板所有檢測的機台來負責該批 IC 板檢測工作（即便該機台正在進行其它批 IC 板的檢測工作或有其他機台閒置中。例如：如果機台 A 正在檢測其他批 IC，機台 B 閒置中，但是如果等機台 A 做完手上的工作後再來檢測該批 IC 的結束時間比由機台 B 來檢測該批 IC 的結束時間早的話，則會把該批 IC 板的檢測工作分配到機台 A 上。完成時間是以秒爲單位，不足一秒無條件進位），如果同時有兩部以上的機台符合上述條件，則該批檢測工作會被分配到速度較快的機台。請寫一個程式來計算最後一批完成檢測工作的起始時間點及機台號碼。

條件限制

1. 「奇上」公司擁有n （$n \le 20$） 台編號爲 1, 2, …, n的IC板檢測機。每台機器檢測的速度分別爲每分鐘能檢測s_1, s_2, …, s_n片IC板 ($1 \le s_1, s_2, \ldots, s_n \le 200$, $s_1 \ne s_2 \ne \ldots \ne s_{n-1} \ne s_n$）。
2. 每台檢測機在開始檢測一批新的 IC 板前需要 5 分鐘的時間將該批 IC 板準備就緒，而在完成一批檢測工作後，也需要 10 分鐘的時間進行自動清理的工作。這些動作一定只能從每分鐘的 0 秒開始進行。換句話說，如果有一批檢測工作在第 23 分鐘又 30 秒時完成，則下一批等待中的 IC 板檢測工作只能從第 39(24+10+5)分鐘開始進行。（請注意！起始時間爲第 39 分鐘而不是第 34 分鐘）

輸入說明：

輸入檔第一行只有一個正整數n，代表「奇上」公司IC板檢測機台數。第二行有n個以空格分開的正整數，分別代表每一機台的檢測速度（即$s_1, s_2, \ldots, s_n$）。接下來的數行每一行有兩個正整數，分別代表每台小貨車抵達「奇上」公司的時間（以分鐘計算）（≤50000）及所需檢測的IC板數量(≤1000)。其中每一行的第一個數字爲非遞減(non-decreasing)（若抵達時間相同則依input順序處理）。此外，最後一行以兩個 0 來代表檔案結束（不要將其當作送檢資料處理）。（所有輸入都保證最後一批檢測工作的起始時間都在 2^{30}內）

輸出說明：

請輸出最後一批完成檢測工作的起始時間點及該機台號碼，這兩個整數應以一個空格分開。輸入保證不會有兩批或兩批以上的工作在最後同時完成。

輸入範例

```
2
10 20
5 100
10 200
15 300
19 500
0 0
```

輸出範例

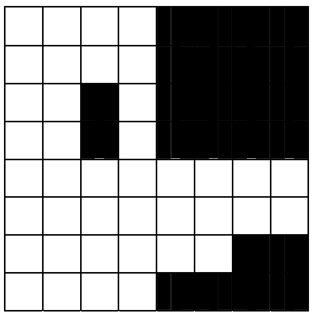
```
60 2
```

3. 黑白影像的四分樹表示法

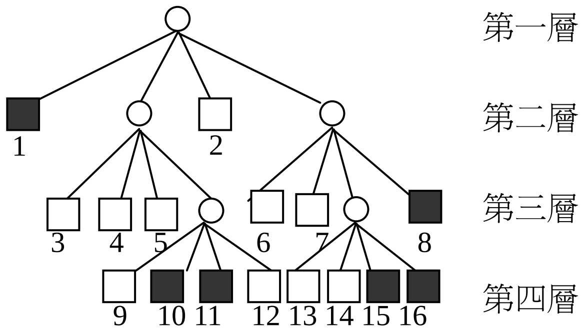
問題敘述：

一張黑白影像可以以一個二維陣列表示，將陣列的每個元素以一個位元表示其顏色，1 代表黑色，0 代表白色。然而在大多數情況下，我們發現一張黑白影像的中，相同的顏色通常會位在相鄰的位置；因此我們可以利用四分樹(quadtree)的表示法來記錄一張黑白影像，以得到比二維陣列較節省記憶體的表示法。

四分樹的表示法就是將目前欲表示的黑白影像區域，以十字形方式分割為四份，分別為東北方、西北方、西南方及東南方四個子區域。以圖一為例，當把整張大小為 16×16 的圖分割為四份時，東北方的恰好為一整塊的黑色，我們可以一個黑色的節點(圖二中的 1 號節點)表示這整個 4×4 的黑色區域。而西北方的 4×4 區域並非由同一色所組成，因此必須再進行分割；當我們以相同的方式遞迴地分割西北方的 4×4 區域，可得到圖二中第 3、4、5 號節點以及再下一層的 9、10、11、12 號節點。圖二中，黑色的節點表示黑色的像素，白色的節點表示白色的像素，圓圈表示內部節點。



圖一



圖二

當我們得到圖二的四分樹表示法後，可使用 b, w, g 三個符號分別代表黑色節點、白色節點及內部節點，依照深先(depth-first)表示法將四分樹符號輸出(依 pre-order 順序輸出)。因此，圖二的四分樹可表示成

```
g b g w w w g w b b w w g w w g w w b b b
```

請撰寫一個程式，將每張黑白影像的四分樹表示法，利用中 b, w, g 三個符號以深先表示法的結果列出。

輸入說明：

第一行爲一個整數*n*，代表一張正方形黑白影像的寬度及高度。($n=2^k$，*k*爲正整數，$1<k\le7$)。

接下來的 *n* 行代表影像中每一行的內容：0 表示一個白色像素，1 表示一個黑色像素。

輸出說明：

對每一個輸入的黑白影像，將其四分樹表示法以 b, w, g 三個符號，依深先表示法順序列出(每個輸出符號之間有一個空白隔開)。

輸入範例 1：

```
4
0000
0100
0011
0011
```

輸出範例 1：

```
g w g w w w b w b
```

輸入範例 2：

```
4
1000
0110
0110
0000
```

輸出範例 2：

```
g g w w b w g w b w b g b w w w g w b w w
```

輸入範例 3：

```
8
```

```
00001111
00001111
00101111
00101111
00000000
00000000
00000011
00001111
```

輸出範例 **3**：

```
g b g w w w g w b b w w g w w g w w b b b
```

4. 經濟編碼

問題敘述：

為降低資料儲存的空間或增加資料傳送的速度，編碼是常用的方法。

假設有一個字元集，每個字元出現的頻率是已知的。現在要把每個字元編碼成為一個二元字串(例如把'A'編碼作 101)，採用的編碼必須合乎以下條件：一個字元的編碼不可以是另一個字元的前置(prefix)。前置的定義如下：若一個字串 s_1 為另一個字串 s_2 的前置，則從 s_2 的最後一個字元開始，連續刪除一定數量的字元後可以得到 s_1 (s_2 本身也是 s_2 的前置)，舉例而言：如果字元'A'的編碼是 101，而字元'B'的編碼為 01，則'B'的編碼不為'A'編碼的前置；如果字元'C'的編碼為 1100，而字元'D'的是 11，則'D'的編碼是'C'編碼的前置。以下的編碼方式可以在符合這個條件下給出最經濟的編碼，請找出使用下述方法做最經濟編碼時，一個字元編碼的預期長度。

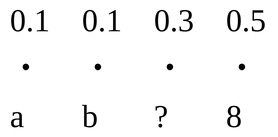
編碼法

1.如以下所述建立一棵二元樹。

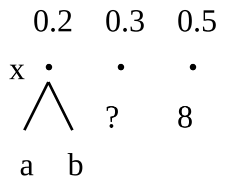
先從字元集選取兩個出現頻率最低的字元作合併，合併後以一個全新的虛擬字元取代這兩個字元，新字元的頻率等於這兩個舊字元頻率的總和，並令這兩個舊字元為此新字元的兩個子樹，左右不拘。重覆以上動作，直至字元集剩下一個字元為止。如下圖(i)到(iv)。

2.再依照以下所述方法將各字元作編碼。

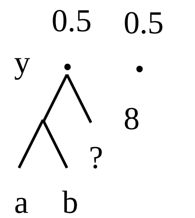
由上一步驟所得之二元樹，將每個內部節點(internal node)連往左子樹的邊(edge)標記為'0'，連往右子樹的邊標記為'1'，如下圖(v)所示。一字元的編碼即為從樹根(root)至此字元，經過的每一個邊的標記所成之字串。在此'a'編碼作 000，'?'編碼作 01。



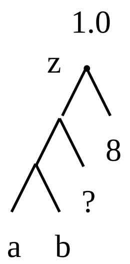
(i) 起始

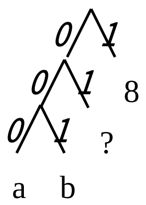


(ii) 合併頻率最低的 a 和 b，並以新字元 x 取代



(iii) 合併頻率最低的 x 和?，並以新字元 y 取代





(iv) 合併 y 與 8，並以 z 取代，完成

(v) 最經濟編碼，最後一個虛擬字元(在此爲 z)，即爲此二元樹的樹根

二元樹建立的步驟。

在按照上述的編碼法完成最經濟編碼之後，就可以計算這個字元編碼的預期長度。首先算出每個字元的預期長度 = 編碼長度 × 出現頻率，然後把所有字元的預期長度總合起來，就可以得到此字元編碼的預期長度。下表是上述編碼的計算範例。

字元	編碼	編碼長度	出現頻率	預期長度
a	000	3	0.1	0.3
b	001	3	0.1	0.3
?	01	2	0.3	0.6
8	1	1	0.5	0.5
	字元編碼的預期長度			1.7

輸入說明：

第一行爲一整數 *n*，代表字元集的大小(0<*n*≤200)。
然後每一個字元分行列出，每行列出一字元外，並列出它出現的頻率，字元與頻率之間以一個空白分隔。(所有字元頻率的總和等於 1)

輸出說明：

預期一個字元編碼的長度。精確至小數點後兩位，小數點後第三位四捨五入。

輸入範例 1：

```
4
a 0.1
b 0.1
? 0.3
8 0.5
```

輸出範例 1：

```
1.70
```

輸入範例 2：

```
6
* 0.3
b 0.3
< 0.05
H 0.25
( 0.05
h 0.05
```

輸出範例 2：

```
2.25
```

5. 高中運動會

問題敘述：

夢幻城市每年爲全市高中生舉辦一次運動大會。爲促進各校同學之間的交流，採用特別的分隊方式：每一個學校的同學，必須被均勻分散到各隊，使得每一隊中該校的人數皆相同。爲增加比賽的競爭性，希望分成越多隊越好。你的任務是由各校的人數，決定最多可分成的隊數。

輸入說明：

輸入檔第一行爲一個介於 1 到 500 間的正整數 *N*，代表學校的個數。皆下來有 *N* 行，每行爲一個介於 1 到 10000 間的正整數，分別代表這 *N* 個學校的人數。

輸出說明：

最多可分成的隊數。

輸入範例 1：

```
3
12
16
20
```

輸出範例 1：

```
4
```

輸入範例 2：

```
4
400
200
150
625
```

輸出範例 2：

6. 線性系統求解

問題敘述：

1947 年美國勞動統計局花了兩年的時間，收集超過 250,000 經濟資訊。兩年後哈佛大學 Wassily Leontief 教授將這些經濟活動分析成 500 產業，其中包括煤業、汽車業、運輸業等。對每一個產業，Leontief 教授用一個線性方程式來表達這個產業如何將其產出分散或影響其他的產業。他的研究開啓了日後利用電腦來解決大型數學問題的先例。Leontief 教授後來在 1973 年獲得諾貝爾經濟貢獻獎。

由於在科學或經濟上所包含的資料都相當多，有很多數學模式都是線性的，所以線性系統求解可以說是相當重要。有關線性系統的解法可以應用高斯消去法 (Gauss-Jordan Elimination) 來求解。

假設我們有以下的線性方程組：

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = c_1$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = c_2$

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = c_3$

可以用消去法來求解：

步驟一：

$x_1 + (a_{12}/a_{11})x_2 + (a_{13}/a_{11})x_3 = c_1/a_{11}$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = c_2$

$a_{32}x_2 + a_{31}x_1 + a_{33}x_3 = c_3$

步驟二：

$x_1 + (a_{12}/a_{11})x_2 + (a_{13}/a_{11})x_3 = c_1/a_{11}$

$(a_{22}-a_{21}*a_{12}/a_{11})x_2 + (a_{23}-a_{21}*a_{13}/a_{11})x_3 = c_2-a_{21}*c_1/a_{11}$

$(a_{32}-a_{31}*a_{12}/a_{11})x_2 + (a_{33}-a_{31}*a_{13}/a_{11})x_3 = c_3-a_{31}*c_1/a_{11}$

接著再對第二行做步驟一，以此類推。

舉例來說，我們現在來解一個方程組：

$2x_1 + 8x_2 + 4x_3 = 2$

$2x_1 + 5x_2 + x_3 = 5$

$4x_1 + 10x_2 - x_3 = 1$

這個方程組可以表示成：

```
2  8  4  2
2  5  1  5
4 10 -1 1
```

解此方程組的步驟如下：

步驟一：

```
1  4  2  1
2  5  1  5
4 10 -1 1
```

步驟二：

```
1  4  2  1
0 -3 -3 3
0 -6 -9 -3
```

步驟三：

```
1  4  2  1
0  1  1  -1
0 -6 -9 -3
```

步驟四：

```
1  4  2  1
0  1  1 -1
0  0 -3 -9
```

步驟五：

```
1  4  2  1
0  1  1 -1
0  0  1  3
```

步驟六：

```
1  4  0 -5
0  1  0 -4
0  0  1  3
```

步驟七：

```
1  0  0  11
```

0 1 0 -4
0 0 1 3

最後可得解爲：
$x_1 = 11$
$x_2 = -4$
$x_3 = 3$

我們接下來討論下面這個方程組：
$2x_1 + 2x_2 + 2x_3 = 2$
$4x_1 + 4x_2 + 4x_3 = 4$
$16x_1 + 16x_2 + 16x_3 = 16$

步驟一：
2   2 2 2
4   4 4 4
16 16 16 16

步驟二：
1 1 1 1
0 0 0 0
0 0 0 0

最後可得解爲：
無限多組解。

然而在計算的過程中，因爲可能使用浮點數的關係，會導致進位誤差以及計算誤差；或者是在不同的計算順序下，也可能會有誤差的產生。如果我們使用分數來表示數字就可以預防浮點數的誤差。我們可以用 1/3 來表示一個數字，取代原本的 0.333333（有遺失精確度的問題）。這麼一來，就可以解決上面所提的錯誤的問題了。

現在請設計一個程式，可以幫我們求解線性系統。注意方程組可能有一組解、無解、或無限多組解。

輸入說明：

爲方便輸入，方程組的變數均省略。第一行是整數 n ($0<n<50$) 代表方程組中有 n

個變數。接下來會有m行（$1\le m\le n$），每一行代表一個方程式，若有n個變數，則一行會有n+1 個值（例：2 8 4 2 代表 $2x_1 + 8x_2 + 4x_3 = 2$），每兩個值之間會有一個或多個空白隔開。方程式中的數字都是整數。

輸出說明：

第一行為 0 或 1 或N分別表示無解、1 組解、及無限多組解。若為 1 組解或無限多組解，請再輸出N行代表一組解，每一行代表一個變數，變數編號從x_1, x_2, 到x_n(無限多組解時，請列出任一組解)。

如果某些變數的解不是整數，請用最簡分數來表示那些解。你可以假設所有數字的分子與分母在運算過程中都不會超過 32-bit integer 的範圍（但要適時地約分)。如果數字為負的分數 p/q，其中 p 或 q 其中一個數字為負數，請將分母修正成正數，分子修正成負數。例：-1/3 是一個合法的表示法，1/-3 則不是。（輸出解答時，只需在等號前面後面各加一個空白，其他地方不用）

輸入範例 1:

```
3
2   8   4   2
2   5   1   5
4  10  -1   1
```

輸出範例 1:

```
1
x1 = 11
x2 = -4
x3 = 3
```

輸入範例 2:

```
3
1   2   3   0
8  10  12   6
7   8   9   6
```

輸出範例 2:

```
N
x1 = 2
x2 = -1
x3 = 0
```

輸入範例 3:

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

輸出範例 3:

```
0
```

輸入範例 4:

```
1
3 10
```

輸出範例 4:

```
1
x1 = 10/3
```