

一一〇學年度高級中學資訊學科能力競賽

台中區複賽程式設計試題

一、五倍卷購物活動 (10 分)

為了配合五倍卷活動，A 店家正在舉辦購物多買多送活動。口罩每盒 X 元，每買一盒口罩送集點卡一點，累積 Y 點就可再換一盒口罩，換到一盒口罩後也可以得到一點，直到無法兌換為止。請你寫一個程式，計算總金額 Z 最多可以買到以及加換到幾盒口罩？還剩下多少元沒有買？剩下多少點沒有兌換？

輸入說明：

輸入資料含多組測試案例。每組測試案例為三個正整數 X , Y , Z ，中間以空格區分，代表口罩每盒 X 元、累積 Y 點就可再換一盒口罩與總金額 Z 。若 X , Z 任一數字為 0 或負數，或者 Y 為 0, 1 或負數，則代表所有測試案例結束。

輸出說明：

每組測試案例輸出三個整數，第一個為總共買到以及加換到幾盒口罩，第二個為剩下多少元沒有買，第三個為剩下沒有兌換的點數，三個數字之間空一個空格。每組測試案例輸出一行。

範例輸入：

100 10 5000

1 10 5000

4 10 5000

0 1 1

範例輸出：

55 0 5

5555 0 5

1388 0 8

二、找出所有可能的 N 值中的質數個數 (10 分)

為了產生一個正整數 N，N 的位數最多為 10。給定多個數對 (X, Y)，其中 X 與 Y 的範圍是 $0 \leq X, Y \leq 9$ (即 0 或正整數)，X 是 N 中某個位數的值，Y 是 X 在 N 中重複出現的次數，但這 Y 個 X 可以出現在 N 的任意位數。舉例 N 由 3 1 2 2 形成，所有 N 可能的數值是 223, 232, 322。接著計算所有可能的 N 值中有幾個質數。

輸入說明：

輸入的每一行的每兩個數字是一個數對 (X, Y)，以空格隔開。若 X, Y 任一數字為負數，代表程式結束。

輸出說明：

輸出的每一行為根據輸入列產生的所有可能的 N 值中有幾個質數。

範例輸入：

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

範例輸出：

```
1
0
1
```

三、地球暖化效應(10 分)

全球暖化，海洋上升，將會淹沒部分陸地。有一四周環海的島嶼，島內各處陸地的海平面高度都不一，居民熱衷製造溫室氣體，導致海平面上升，當海平面上升的高度大於等於陸地原來的海平面高度時，陸地就會被海水淹沒，如此 1 個島嶼就有可能被分割成 2 個甚至多個島嶼。

若以 $m \times n$ 矩陣中的元素代表相對應陸地的海平面高度，矩陣之外代表海洋，例如：

有一 5×5 矩陣 $A = \begin{bmatrix} 3 & 4 & 3 & 0 & 0 \\ 3 & 5 & 5 & 4 & 3 \\ 2 & 5 & 4 & 4 & 3 \\ 1 & 3 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{bmatrix}$ ，可看出只有框起來的地方被海平面吞沒，島嶼還沒被分隔

開，即使海平面升高為 5，整個島嶼都被淹沒了，仍不會被分隔成 2 個甚至多個島嶼。

另一 5×5 矩陣 $B = \begin{bmatrix} 5 & 5 & 5 & 5 & 7 \\ 4 & 1 & 1 & 1 & 4 \\ 4 & 1 & 2 & 1 & 3 \\ 7 & 1 & 0 & 0 & 4 \\ 7 & 3 & 3 & 4 & 4 \end{bmatrix}$ ，當海平面上升 3 時，可從下圖 B' 看出島嶼中的陸地被框線

切分開，分離成左上與右下 2 塊島嶼，

$B' = \begin{bmatrix} 5 & 5 & 5 & 5 & 7 \\ 4 & 1 & 1 & 1 & 4 \\ 4 & 1 & 2 & 1 & 3 \\ 7 & 1 & 0 & 0 & 4 \\ 7 & 3 & 3 & 4 & 4 \end{bmatrix}$

今給定一個整數矩陣，給數個元素海拔高度的島嶼，海洋必須上升多高才會使陸地分裂成碎片？只有同一列或同一行的元素才視為相鄰，若某一元素上下左右相鄰 4 元素均為 0，則可視為該元素周圍將被海洋包圍；島嶼內之陸地地面，若存在有比海平面低的窪地，而此窪地周圍仍環繞陸地時，海水是不會流入造成分隔。

輸入說明：

可連續輸入多筆測資。

每個測資第 1 行包含兩個正整數 m n ，表示矩陣的維度， m n 兩者都不會超過 100。

接著輸入 m 行，每行包含 n 個正整數；正整數表示網格元素的原始高度。假設海洋最初圍繞給定的網格。該島最初是相連的。每個高度永遠不會超過 1000。

最後一行測資是“0 0”。

輸出說明：

對於每個測資，輸出：

“NEVER SPLIT”（島嶼淹沒都不會分裂）或

“SPLIT WHEN OCEAN RISES 4”（當海平面上升 4，島嶼就會分裂）

範例輸入：

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

範例輸出：

```
NEVER SPLIT
SPLIT WHEN OCEAN RISES 3
```

四、檔案編碼(10 分)

定義一個編碼演算法，利用數字編碼的字典來取代原有的英文字母，說明如下：

首先統計所有出現的英文字母依序編號，假如 26 個英文字母都出現在檔案中，則最初編碼字典即為 (A, 00), (B, 01), (C, 02), ..., (Z, 25)，也就是英文字母 A 用 2 位數字 00 取代，英文字母 B 用 2 位數字 01 取代，英文字母 C 用 2 位數字 02 取代...，英文字母 Z 用 2 位數字 25 取代。如果我們將一個文字檔案重新編碼，假設檔案中只出現 A、B、C、D，4 種英文字母，我們可以編碼 (A, 0)、(B, 1)、(C, 2) 和 (D, 3)，也就是英文字母 A 用 1 位數字 0 取代，英文字母 B 用 1 位數字 1 取代，英文字母 C 用 1 位數字 2 取代，英文字母 D 用 1 位數字 3 取代。那麼到底要用幾位數字編碼，要看字典所用的英文字母的總數量來決定。

演算法進行如下：

1. 找到尚未編碼的領先字母串部分，尋找在字典中出現最長相符的字母串，並使用該字母串所表示的數字取代。
2. 如果還沒有編碼完畢，則在字典中新增一個新的元素(s, n)，其中 s = 剛剛被編碼的字母串加上它隨後的一個字母，形成新的字典元素，n=字典中尚未使用的最小數字。
3. 欲使用數字 n 替換相符的字母串時，若 n 的位數因為增加字典元素而增加時，要替換發生時字典中最長編碼的大小，也就是當新增一個新的元素(s_i, 10)時，字典內原來所有元素的數值 n 須由 1 位數同時擴展為 2 位數，並從此開始，用 2 位數取代將替換的字母串。

舉例說明，假設我們要編碼的原來文字檔內容為 PQPQQPPQQ，預設字典為：(P, 0)和(Q, 1)。下表顯示了編碼文字檔的步驟：

| 檔案內容字串 | 字典中出現最長相符的字母串 | 取代數字 | 新增字典元素 |
|-----------|---------------|------|----------|
| PQPQQPPQQ | P | 0 | (PQ, 2) |
| 0QPQQPPQQ | Q | 1 | (QP, 3) |
| 01PQQPPQQ | PQ | 2 | (PQQ, 4) |
| 012QPPQQ | QP | 3 | (QPP, 5) |
| 0123PQQ | PQQ | 4 | --- |

最終編碼完的結果是 01234。

對於上面的字典，如果要重新編碼的字母串足夠多到使(s_i, 10)的新元素增加到字典中，那麼從這一時點開始，要編碼的字母串中使用的所有數字替換字母串必須擴展到 2 位數字，(即，P 將用 00 取代，Q 將用 01 取代，PQ 將用 02 取代...等)；如果一個新元素 (s_i, 100) 被添加到字典中，從這一點開始的所有替換都將增加到 3 位數，依此類推。

因此，較長的字母串 “PQPQQPPQPPQPPQPQ” 將被編碼為 01234027301，而不是 0123402731。

輸入說明：

每個測試資料將由兩行組成。

第一行為已重新編碼後的數字串，將要解編碼回復原始檔案字母串內容。

第二行將包含編碼中使用的初始字典。這一行將以一個正整數 n 開始 (1 ≤ n ≤ 100)，表示初始字典中的元素個數，然後是 n 個字母字符串。其中第一個將與字典中的 0 配對，第二個與 1 配對，依此類推。

最後一個測試資料一行單個 “0” 表示結束。

輸出說明：

對於每個測試例，輸出一行，每一行是解編碼後的字母串。

範例輸入：

01234

2 A B

01234027301

2 A B

02111518202927

26 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

21104

3 BA A C

01

2 XYZ AB

0

範例輸出：

ABABBAABB

ABABBAABBAABAABAB

CLPSUSULP

CAABAAA

XYZAB

五、置換密碼(10 分)

瓊斯博士收到亨利博士從南非寄來的一封信，信上的內容如下：

親愛的小亨利：

若要打開失落的法櫃，必須把我給你的一段文字內容將其每個文字依照上帝指引之路轉譯成代碼(由 0-F 組成)；存放在我書房中保險櫃的置換密文表，可以協助你將代碼轉換成密文。最後要打開上帝遺失的法櫃，還需要將密文重新排列後才會是正確的密碼，至於排列規則可以從範例獲得靈感。

我相信你是那位領受聖靈印記之人。

亨利一世

上帝指引之路：

- I. 將每個文字的第 0 個位元內容進行 NOT 運算後成為代碼的第 0 位元
- II. 將每個文字的第 1、3、4、5 位元進行 XOR 運算後成為代碼的第 1 位元
- III. 計算每個文字裡的位元內容出現 '1' 的次數是偶數就填入 0，反之就填入 1 於第 2 位元。
- IV. 取出每個文字的第 3 個位元做為代碼的第 3 位元

日記的置換密文表

| 代碼 | 密文 | 代碼 | 密文 | 代碼 | 密文 | 代碼 | 密文 |
|--|-----|----|------|----|------|----|------|
| 0 | \$# | 4 | \$@[| 8 | #<< | C | \$[|
| 1 | [| 5 | !< | 9 | < > | D | @<# |
| 2 | >#[| 6 | # | A |][| E | \$>! |
| 3 | >]] | 7 | ^@! | B | ^^\$ | F | @\$ |
| 密文符號優先權，由最左的最高優先權排至最右的最低優先權： ^] [@ > < \$ # ! | | | | | | | |

輸入說明：

一段文字內容由英文字母、數字和符號組成的字串 L，其字串長度為 $1 \leq L \leq 500$ 。

輸出說明：

根據轉譯後密文內符號出現次數決定該密文符號的輸出順序，若是遇到相同出現次數則依照密文符號優先順序排之。

範例 1 輸入：

Proper preparation solves 80% of life' s problems.

範例 1 輸出：

]][#><!^@\$

範例 2 輸入：

Every noble work is at first impossible.

範例 2 輸出：

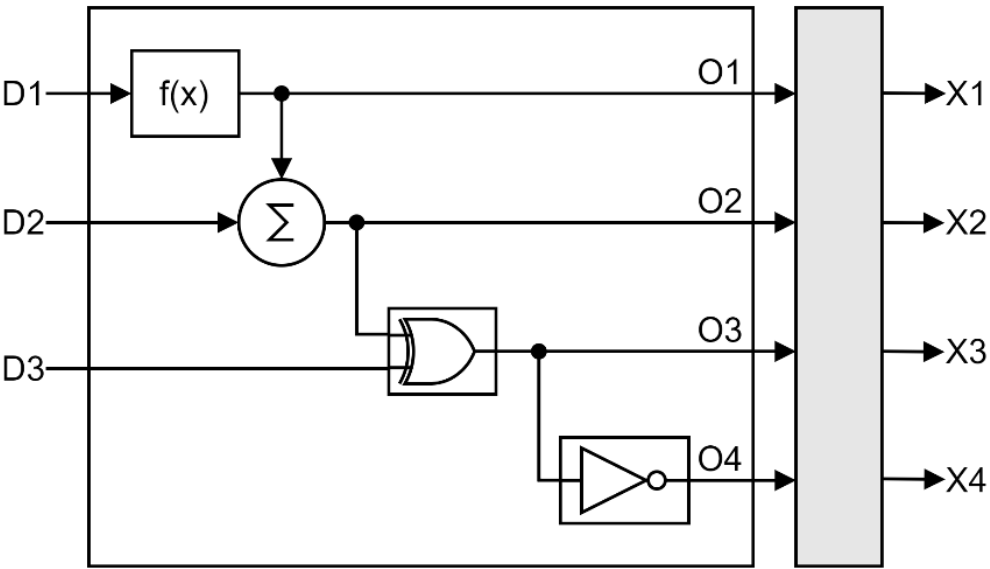
<#[!>|^]@\$

六、進制轉換(10 分)

赫丘勒白羅拿到一個莫里亞蒂寄來的一封信、神秘盒子、數張卡片和一個轉換表。信裡的內容如下：這是給我最尊敬的對手赫丘勒的一個全新挑戰，盒子上的神秘圖案暗示內部的運作機制。這個神秘盒子有 3 個卡片輸入端，分別為 D1、D2 和 D3，用來插入我給你的這些卡片，透過內部運算後產生 4 個輸出結果，分別是 O1、O2、O3 和 O4。最後依據下圖轉換表將 O1、O2、O3 和 O4 分別轉換成 X1、X2、X3 和 X4。

看在多年的交情上，給你一些關於神秘盒子的提示：

- (1) 內部架構仍是遵循電腦的 2 進制運算方式。
- (2) 每筆資料會根據資料長度儲存不會多也不會少。
- (3) 資料長度不一致的資料進行運算時，內部盒子此時才會給予足夠儲存空間進行運算。
- (4) f(x)會依據 D1 的資料長度去計算 D1 儲存空間可允許的最大值。
- (5) 卡片上的 D1、D2 和 D3 採取轉換表的符號 D 組合，而 X1、X2、X3 和 X4 採取轉換表的符號 X 組合。



盒子上的神秘圖案

轉換表可以提供你符號 D 和符號 X 的對應關係，例如符號 D 的「*!」對應符號 X 的「<」。

| 符號 D | 符號 X | 符號 D | 符號 X | 符號 D | 符號 X |
|------|------|------|------|------|------|
| # | * | *^ | @ | ^@ | (|
| * | : | *) | ? | ^(| ; |
| ^ | { | *@ | ` | ^\$ | + |
|) | ” | *(| > | ^& | = |
| @ | [| *\$ | , | ^! |] |
| (| ~ | *& | \ | ^% | _ |
| \$ | & | *! | < |)# | ! |
| & | } | *% | ^ |)* | # |
| ! |) | ^# | , |)^ | :* |
| % | . | ^* | |)) | :: |
| *# | \$ | ^^ | % |)@ | :{ |
| ** | - | ^) | / |)(| :” |

為了幫助赫丘勒，請你試著撰寫 1 個程式去分別讀取 D1、D2 和 D3 的符號 D 組合，依據祕密盒子的內部運算機制後，分別輸出 X1、X2、X3 和 X4 的符號 X 組合。

輸入說明：

第 1 行為 1 個整數，代表接下來共有 M 組測試資料，其中 $1 \leq M \leq 5$ 。

每組測試資料包含 3 筆 D 符號組合的輸入，分別為 D1、D2 和 D3，其範圍為 $\# \leq D1, D2, D3 \leq *!@\$ \& @ \# \&) \& \# \% ((* \$ * ($

輸出說明：

根據每組測試資料中的 D1, D2 和 D3，經由神秘盒子的運算處理，請分行輸出 X1、X2、X3 和 X4 等 X 符號的符號組合。

範例輸入：

```
2
!!$
)^(*
*%$(
(&)(
$@^&)
(&@)@%
```

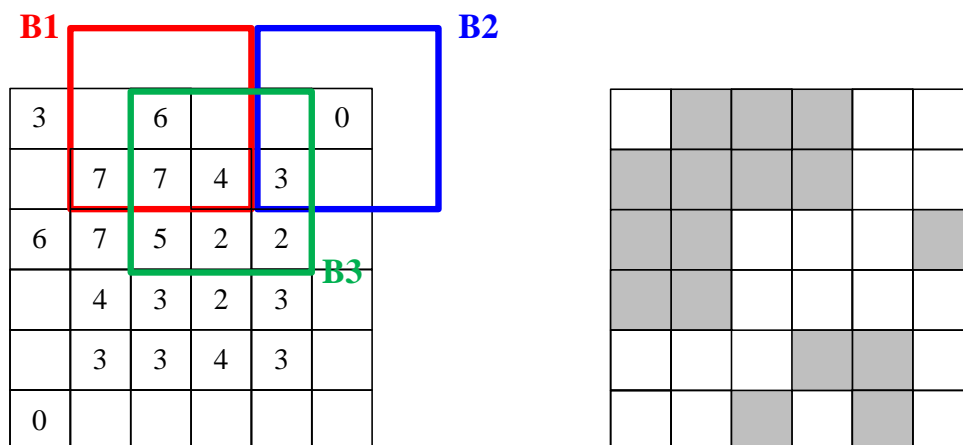
範例輸出：

```
##
[~<
~(#
{*
}##
{&('
^%[_
@.= {
```

七、數字地雷(10 分)

某片正方形荒地由 $N \times N$ 個區塊組成，每一區塊都可能埋有地雷，請你找出所有埋地雷的區塊，進而計算出橫向各列每一列的地雷總數。找地雷的唯一線索是區塊上可能存在的數字，若某一個區塊 p 上有一個數字 M ，則表示以 p 為中心的 3×3 區塊總共有 M 個區塊埋有地雷。

舉例來說，下方左圖為一個 6×6 區塊所形成的荒地，左上方紅色粗線框形成一個 3×3 區塊 $B1$ ，但其中有 3 個區塊超出範圍是不存在的。因 $B1$ 的中心區塊數字為 6，表示 $B1$ 內的地雷總數為 6。同樣地，對於右上方藍色粗線框所形成的 3×3 區塊 $B2$ ，其中心的區塊數字為 0，表示 $B2$ 內的地雷總數為 0，接著便可再找到下方綠色粗線框 $B3$ 內的地雷分布情況。依此方式，最後即可找出下方右圖整個 6×6 區塊之地雷分布狀況，灰色區塊為地雷所在區域。



輸入說明：

第 1 行有 1 個數字 N ，表示整個荒地有 $N \times N$ 個區塊。

第 2 ~ $(N+1)$ 行，每一行有 N 個符號，若符號是數字(0~9)，則為對應區塊上的數字。反之若符號是 '.'，則表示對應的區塊沒有數字。

輸出說明：

有 N 個數字，表示 $N \times N$ 個區塊中橫向 N 列由上至下每一列的地雷總數。

範例輸入：

```
6
3.6..0
.7743.
67522.
.4323.
.3343.
0.....
```

範例輸出：

```
3 4 3 2 2 2
```

八、燈泡亮燈組合(10 分)

房間內有 M 個開關及 N 個燈泡，每一個開關可控制若干個燈泡之亮及暗。按 1 次開關，其所控制之燈泡若原為暗的狀態就會轉為亮，反之，若原為亮的狀態就會變暗。所有燈泡之初始狀態均為暗，在給定某些燈泡要為亮(其餘燈泡為暗)的目標下，請找出所按開關次數最少，且開關號碼最小之情況，依開關編號由小至大印出這些號碼。所使用之開關次數限制最多為 3 次，若在 3 次內找不到答案則印出 X。

假設現有 6 個開關(編號 1~6)及 6 個燈泡(編號 1~6)，下表是每個開關所控制的燈泡號碼。假設我們的目標是要讓編號 1、5、6 的燈泡變亮，我們可以先按下開關 3 讓燈泡 3、4、5 變亮，接著按開關 4 讓燈泡 1、2 變亮。最後按開關 6 再讓燈泡 6 變亮，同時讓燈泡 2、3、4 變暗，因此我們需要按 3 次開關 3、4、6。

| 開關號碼 | 控制之燈泡號碼 |
|------|------------|
| 1 | 3, 5 |
| 2 | 2, 4, 5 |
| 3 | 3, 4, 5 |
| 4 | 1, 2 |
| 5 | 1 |
| 6 | 2, 3, 4, 6 |

輸入說明：

第 1 行有 2 個數字 M 及 N ，分別表示開關數及燈泡數。

第 2 ~ ($M+1$)行分別表示每個開關所控制的燈泡號碼。在第 i 行中，所列的即為開關編號 $i-1$ 所控制的所有燈泡號碼 (1~ N)。

第 $M+2$ 行為最後要亮燈的所有燈泡號碼 (其餘燈泡為暗)。

輸出說明：

若干個數字，代表所使用的最少數、且號碼最小之開關號碼 (由小至大排序)，最多 3 個數字，若無解則印出 X。

範例輸入：

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

範例輸出：

```
3 4 6
```

九、最大工作衝刺期(10 分)

老陳是公司部門的主管，最近為了解相關專案進度與部門內部人員工時，因此向資訊部門取得底下員工的工時紀錄表，以便調整專案進度與員工加班情形，由於員工眾多且時數紀錄繁瑣，所以決定撰寫一隻程式檢查。由於公司性質關係，工作時數較為彈性，正常日的工作時數不超過 8 小時，加班日是工時超過 8 小時。為了更好預估專案與工時情況，老陳定義了工作衝刺期，統計所有超時日大於正常日的區段，並印出最大區段的長度。

請幫老陳完成程式，算出最大工作衝刺期區間長度。

舉例：

若過去一段工時紀錄 $X = [0\ 5\ 6\ 7\ 8\ 9\ 9\ 10\ 3\ 8\ 7]$ ，檢查的區段則為

$[0]$

$[0\ 5]$

$[0\ 5\ 6]$

...

$[5]$

$[5\ 6]$

$[5\ 6\ 7]$

...

$[0\ 5\ 6\ 7\ 8\ 9\ 9\ 10\ 3\ 8\ 7]$

輸出結果為

5（意即最大的工作衝刺期長度為 5，如 $[7\ 8\ 9\ 9\ 10]$ 、 $[9\ 9\ 10\ 3\ 8]$ ）

輸入說明：

輸入資料包含多個數字，數字間以空白區隔，每個數字代表一個工作時數，-1 代表資料輸入結束。

輸出說明：

輸出最大工作衝刺期長度。

範例 1 輸入：

9 9 10 -1

範例 1 輸出：

3

範例 2 輸入：

0 5 6 7 8 9 9 10 3 8 7 -1

範例 2 輸出：

5

範例 3 輸入：

7 8 6 7 3 1 1 2 11 1 3 14 2 4 16 4 8 1 6 13 -1

範例 3 輸出：

1

十、蓄水量計算(10 分)

阿成是國外水資源相關的分析師，由於天氣的變化使得往年水庫穩定的蓄水量起了很大的變化，為了瞭解天氣問題，阿成想藉由蓄水情形來推斷未來天氣可能變化，因此與當地機關申請資料。申請的資料內每一個筆資料具有三個基本欄位：日期時間、降雨量及用水量，每個欄位都以'逗號'分開，顯示如下：

```
2001-01-01-00, 6, 1
2001-01-01-01, 8, 1
2001-01-01-02, 5, 1
...
2001-01-01-21, 2, 15
2001-01-01-22, 0, 1
2001-01-01-23, 0, 1
```

根據阿成的經驗，統計是衡量的一個方法，所以採用以下的方式作為計算蓄水量的標準：

最終蓄水量 = 起始蓄水量 + 當日平均降雨量 - 當日最大用水量

由於阿成不會寫程式，請幫阿成完成程式，使用申請的資料、起始蓄水量、以及上述方法計算每日最終蓄水量，為了計算方便，平均降雨量要四捨五入到整數。

輸入說明：

第一行為一個整數，代表起始蓄水量。

接下來每一行為一筆資料， -1 代表資料輸入結束。

輸出說明：

若最終蓄水量>100，輸出以下資料：

該日最後一小時 當日平均降雨量 當日最大用水量 最終蓄水量

否則，輸出以下資料：

該日最後一小時 不足水量

(不足水量 = 最終蓄水量 - 100)

範例輸入 1：

```
1234
2001-01-01-00, 6, 1
2001-01-01-01, 8, 1
2001-01-01-02, 5, 1
2001-01-01-03, 5, 3
2001-01-01-04, 5, 4
2001-01-01-05, 25, 5
2001-01-01-06, 0, 8
2001-01-01-07, 0, 7
```

2001-01-01-08, 0, 10
2001-01-01-09, 0, 8
2001-01-01-10, 1, 15
2001-01-01-11, 0, 9
2001-01-01-12, 0, 3
2001-01-01-13, 2, 3
2001-01-01-14, 24, 3
2001-01-01-15, 2, 3
2001-01-01-16, 2, 10
2001-01-01-17, 3, 3
2001-01-01-18, 0, 20
2001-01-01-19, 0, 17
2001-01-01-20, 2, 18
2001-01-01-21, 2, 15
2001-01-01-22, 2, 1
2001-01-01-23, 2, 1

-1

範例輸出 1：

2001-01-01-23 4 20 1218

範例輸入 2：

1000

2001-01-01-00, 6, 1
2001-01-01-01, 8, 1
2001-01-01-02, 5, 1
2001-01-01-03, 5, 3
2001-01-01-04, 5, 4
2001-01-01-05, 25, 5
2001-01-01-06, 0, 8
2001-01-01-07, 0, 7
2001-01-01-08, 0, 10
2001-01-01-09, 0, 8
2001-01-01-10, 1, 15
2001-01-01-11, 0, 9
2001-01-01-12, 0, 3
2001-01-01-13, 2, 3
2001-01-01-14, 24, 3
2001-01-01-15, 2, 3
2001-01-01-16, 2, 10
2001-01-01-17, 3, 3
2001-01-01-18, 0, 20
2001-01-01-19, 0, 17

2001-01-01-20, 2, 18
2001-01-01-21, 2, 15
2001-01-01-22, 2, 1
2001-01-01-23, 2, 1
-1

範例輸出 2：

2001-01-01-23 4 20 984

範例輸入 3：

500
2001-01-01-00, 6, 1
2001-01-01-01, 8, 1
2001-01-01-02, 5, 1
2001-01-01-03, 5, 3
2001-01-01-04, 5, 4
2001-01-01-05, 25, 5
2001-01-01-06, 0, 8
2001-01-01-07, 0, 7
2001-01-01-08, 0, 10
2001-01-01-09, 0, 8
2001-01-01-10, 1, 15
2001-01-01-11, 0, 9
2001-01-01-12, 0, 3
2001-01-01-13, 2, 3
2001-01-01-14, 24, 3
2001-01-01-15, 2, 3
2001-01-01-16, 2, 10
2001-01-01-17, 3, 3
2001-01-01-18, 0, 20
2001-01-01-19, 0, 17
2001-01-01-20, 2, 18
2001-01-01-21, 2, 15
2001-01-01-22, 2, 1
2001-01-01-23, 2, 1
2001-01-01-00, 6, 1
2001-01-02-01, 0, 1
2001-01-02-02, 0, 1
2001-01-02-03, 0, 3
2001-01-02-04, 0, 4
2001-01-02-05, 0, 5
2001-01-02-06, 0, 30
2001-01-02-07, 0, 7
2001-01-02-08, 0, 10

2001-01-02-09, 0, 8
2001-01-02-10, 0, 15
2001-01-02-11, 0, 9
2001-01-02-12, 0, 3
2001-01-02-13, 0, 3
2001-01-02-14, 0, 3
2001-01-02-15, 0, 3
2001-01-02-16, 0, 100
2001-01-02-17, 0, 3
2001-01-02-18, 0, 20
2001-01-02-19, 0, 17
2001-01-02-20, 0, 18
2001-01-02-21, 0, 15
2001-01-02-22, 0, 1
2001-01-02-23, 0, 1
2001-01-03-00, 0, 1
2001-01-03-01, 0, 1
2001-01-03-02, 0, 1
2001-01-03-03, 0, 3
2001-01-03-04, 0, 4
2001-01-03-05, 0, 5
2001-01-03-06, 0, 8
2001-01-03-07, 0, 7
2001-01-03-08, 0, 10
2001-01-03-09, 0, 8
2001-01-03-10, 0, 15
2001-01-03-11, 0, 9
2001-01-03-12, 0, 3
2001-01-03-13, 0, 3
2001-01-03-14, 0, 3
2001-01-03-15, 0, 3
2001-01-03-16, 0, 10
2001-01-03-17, 0, 3
2001-01-03-18, 0, 20
2001-01-03-19, 0, 17
2001-01-03-20, 0, 18
2001-01-03-21, 0, 45
2001-01-03-22, 0, 1
2001-01-03-23, 0, 1
2001-01-04-00, 0, 1
2001-01-04-01, 0, 1
2001-01-04-02, 0, 3
2001-01-04-03, 0, 4

2001-01-04-04, 0, 5
2001-01-04-05, 0, 8
2001-01-04-06, 0, 7
2001-01-04-07, 0, 10
2001-01-04-08, 0, 8
2001-01-04-09, 0, 15
2001-01-04-10, 0, 9
2001-01-04-11, 0, 3
2001-01-04-12, 0, 3
2001-01-04-13, 0, 3
2001-01-04-14, 0, 3
2001-01-04-15, 0, 10
2001-01-04-16, 0, 3
2001-01-04-17, 0, 20
2001-01-04-18, 0, 17
2001-01-04-19, 0, 18
2001-01-04-20, 0, 15
2001-01-04-21, 0, 80
2001-01-04-22, 0, 113
2001-01-04-23, 0, 1
2001-01-05-00, 0, 1
2001-01-05-01, 0, 3
2001-01-05-02, 0, 4
2001-01-05-03, 0, 5
2001-01-05-04, 0, 8
2001-01-05-05, 0, 7
2001-01-05-06, 0, 23
2001-01-05-07, 0, 8
2001-01-05-08, 0, 47
2001-01-05-09, 0, 50
2001-01-05-10, 0, 126
2001-01-05-11, 0, 3
2001-01-05-12, 0, 3
2001-01-05-13, 0, 3
2001-01-05-14, 0, 10
2001-01-05-15, 0, 3
2001-01-05-16, 0, 20
2001-01-05-17, 0, 17
2001-01-05-18, 0, 18
2001-01-05-19, 0, 15
2001-01-05-20, 0, 1
2001-01-05-21, 0, 1
2001-01-05-22, 0, 15

2001-01-05-23, 0, 1

-1

範例輸出 3：

2001-01-01-23 4 20 484

2001-01-02-23 0 100 384

2001-01-03-23 0 45 339

2001-01-04-23 0 113 226

2001-01-05-23 0