

台北市九十六學年度高中資訊學科能力競賽

程式設計試題

<p>准考證號碼: _____</p> <p>姓名: _____</p> <p>USB 號碼: _____</p>

下表為評分用，請勿填寫

題目	積木的拼疊問題 Bricks	錄製專輯 Record	會議中心 Room	序列長度問題 Sequence	樹狀結構展示 Tree
測試一					
測試二					
測試三					
測試四					
測試五					
總分					

總分 _____ / 125

選手簽名 _____

評審 _____

注意事項

- 一、此競賽採部分電腦線上自動評分，因此參賽選手應嚴格遵守每一題目所規定之目錄、執行檔、輸入檔、輸出檔之路徑、名稱及格式。若未遵守，該題將可以 0 分計算。你可以用隨身碟內之 check.exe 程式來檢查目錄、執行檔、輸入檔、輸出檔之路徑及名稱是否正確。

題目	積木的拼疊問題 Bricks	錄製專輯 Record	會議中心 Room	序列長度問題 Sequence	樹狀結構展示 Tree
目錄	C:\bricks	C:\record	C:\room	C:\sequence	C:\tree
可執行檔	C:\bricks\bricks.exe	C:\record\record.exe	C:\room\room.exe	C:\sequence\sequence.exe	C:\tree\tree.exe
輸入檔	C:\bricks\input.txt	C:\record\input.txt	C:\room\input.txt	C:\sequence\input.txt	C:\tree\input.txt
輸出檔	C:\bricks\output.txt	C:\record\output.txt	C:\room\output.txt	C:\sequence\output.txt	C:\tree\output.txt
總分	25	25	25	25	25

- 二、針對每一題所完成之程式，其執行時間以 10 秒為限。執行時間超過 10 秒者視同未完成，該題將以 0 分計算。
- 三、競賽結束後，參賽選手應在指定的時間內與評審共同進程式測試與評審作業。
- 四、競賽隨身碟中有 5 個檔案列舉如下（假設隨身碟磁碟機代號為 D:\），分別為題目以檔案輸出入之範例輸入檔。你可以將他們複製到 C 槽相對應的目錄下，用以測試你的程式輸入格式是否正確。

```
D:\bricks\input.txt
D:\record\input.txt
D:\room\input.txt
D:\sequence\input.txt
D:\tree\input.txt
```

- 五、請將你所寫的程式原始檔複製一份到 USB 隨身碟中。但是評分時，將以電腦上 C 槽相關目錄內之程式為評分依據，USB 隨身碟僅供備份及賽後檢測用。

題目一：積木的拼疊問題 (Bricks)

輸入檔：input.txt

輸出檔：output.txt

問題描述

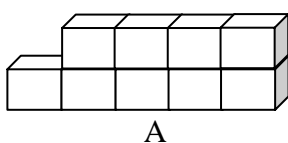
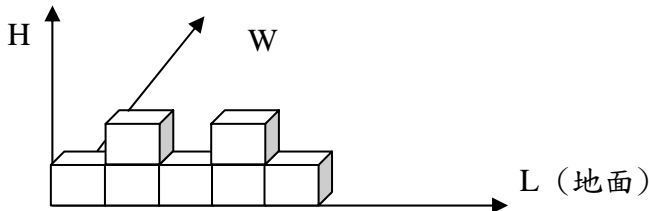
小明和同學一起組隊參加拼疊積木比賽，比賽規則如下：

1. 由各隊隊員在限定時間內到運動場的沙地上尋找小積木。
2. 利用找到的小積木，每二個對應為一組，拼成正方形或長方形的大積木。
3. 看看那一隊找到的小積木所組成的大積木面積最大，即可獲勝。

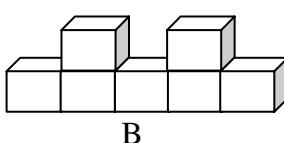
如果我們知道小明這一隊所找到小積木個數與它們的高度和長度，請你寫一個程式算出他們利用這些小積木可以拼出來的大積木的最大面積。

說明：

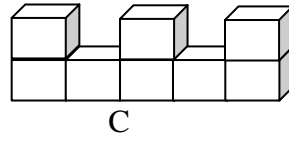
1. 每個埋在沙中的小積木寬度(W)若為一單位，則長度(L)與高度(H)為一單位的整數倍，且長度與寬度的四個邊中至少有一個邊是整齊的，不會有凹凸的狀況。小積木為實心的，沒有挖洞，而且不會是正方形或長方形，下圖所示為一部份小積木的樣子。



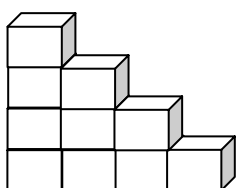
A



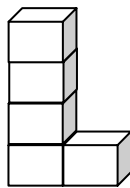
B



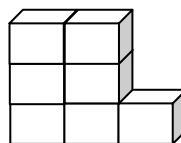
C



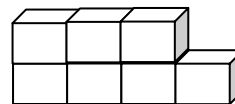
D



E



F



G

2. 若將小積木整齊的那一個邊對齊坐標軸 L ，則上圖小積木皆可用一串數字表示，這一串數字的個數等於小積木的長度，而每一個數字則代表其對應的高度。例如：小積木 $A = (1, 2, 2, 2, 2)$ 、 $B = (1, 2, 1, 2, 1)$ 、 $C = (2, 1, 2, 1, 2)$ 、 $D = (4, 3, 2, 1)$ 、 $E = (4, 1)$ 、 $F = (3, 3, 1)$ 、以及 $G = (2, 2, 2, 1, 1)$ 。
3. 小積木是可以利用旋轉或翻轉來組成正方形或長方形的大積木的，但所組成的大積木必需是實心的，也就是小積木與小積木的接觸面必需密合，中間不能有洞產生。例如上圖中的小積木 B 與 C 可以組成一個 3×5 的長方形大積木，而 E 和 G 則可以組成 3×4 的長方形大積木。另外，如果有二個 E 就可以組成 5×2 的長方形大積木，而二個 D 可以組成 5×4 的正方形大積木了。

條件限制：

1. 小積木的長(L)寬(W)高(H)之範圍如下： $W=1$ 、 $1<L<30$ 、 $1<H<30$ 。
2. 小積木的個數最多 50 個，它們的樣子是可以重覆出現的。
3. 任何一個正方形或長方形的大積木只能由二個小積木組成，但小積木在組合時可以旋轉或翻轉。
4. 若找到的小積木皆無法組成大積木，請輸出 0。

輸入檔格式 (C:\bricks\input.txt)

1. 檔案第一行的數字代表小積木的個數。
2. 檔案第二行以後，每一行皆由一串數字組成，數字間利用空白分隔，這一串數字表示一個小積木，表示方式如說明(2)。

輸出格式 (C:\bricks\output.txt)

輸出的數字為由小積木組成的大積木的最大面積。

輸入檔範例 1

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

輸出範例 1

```
20
```

輸入檔範例 2

```
4
4 1
1 4
1 1 1 2
4 1
```

輸出範例 2

```
10
```

題目二：錄製專輯 (Record)

輸入檔：input.txt

輸出檔：output.txt

問題描述

Jolin 是個愛唱歌的小孩，每次總喜歡邊唱邊用電腦把自己的歌聲錄下來，因此長久下來，在她的電腦裡，已儲存了為數不小的個人歌唱作品。由於耶誕節快要到了，為了準備一份特別的耶誕禮物給爸爸，Jolin 準備從電腦中儲存的個人歌唱作品，挑選幾首歌製成一張個人專輯 CD。由於每張 CD 的容量有限，而 Jolin 的個人歌唱作品早已遠遠超過一張 CD 可收錄的容量，因此 Jolin 希望你可以幫她想辦法，讓她所製作的專輯中，能有數目最多的歌曲（請注意：每一首歌只能被收錄一次），同時必需剛好裝滿整張 CD，不留下任何未使用的空間。

輸入檔格式 (C:\record\input.txt)

輸入檔中的第一行為一個正整數 N ，代表 Jolin 的個人歌唱作品數目。第二行則有 N 個以空白相間隔的正整數 X_i ，分別代表第 i 首個人歌唱作品的大小(單位為 MBytes)。第三行則有一個正整數 S ，代表 CD 的容量(單位為 MBytes)。

為簡化計算過程起見，我們假設每一首歌唱作品的大小皆不相同，同時 $N \leq 100$ ， $X_i \leq 200$ ， $S \leq 10000$ 。

輸出格式 (C:\record\output.txt)

請根據輸入檔的資料，在輸出檔中依序印出兩個正整數 L 與 K 。其中， L 代表最多可以在 CD 中收錄的歌曲數目， K 代表共有幾種方式可以收錄 L 首歌曲於 CD 中。注意，若歌曲曲目相同，但排列順序不同，仍視為不同的收錄方式。

若沒有任何方法可以錄滿整片 CD，則 $L=K=0$ 。

輸入檔範例 1

```
5
10 50 30 70 60
80
```

輸出範例 1

```
2 4
```

輸入檔範例 2

```
5
10 50 30 70 60
20
```

輸出範例 2

```
0 0
```

題目三：會議中心 (Room)

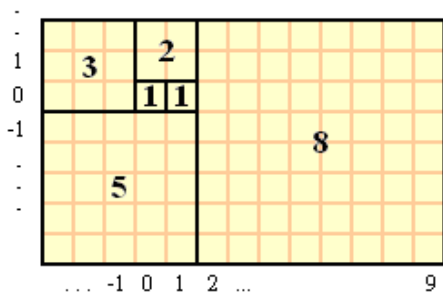
輸入檔：input.txt

輸出檔：output.txt

問題描述

拼拼樂會議中心是一個 $N \times N$ 的超大型可分割式會議中心。每一個 1×1 的空間都可以用隔板隔開，因此該會議中心最多可以有 n^2 個獨立的 1×1 會議室，如要較大的會議室，則需將隔板拿掉使得二或更多個相鄰的 1×1 空間可以合併使用。圖一的會議中心最多可分隔成 169 個 1×1 小會議室，最少則全部合併成為一個 13×13 的會議室。每間 1×1 會議室皆以其二維平面座標為編號。選定一個 1×1 會議室並給予編號 $(0, 0)$ ，相鄰的上、下、左、右會議室編號則依序為 $(0, 1), (0, -1), (-1, 0), (0, 1)$ 。

會議中心外租會議室時，必須按照下列規則，組成合乎需求的會議室。一開始先以編號為 $(0, 0)$ 的空間供租用，如果空間不足，則依序向右方、上方、左方、下方的空間合併成為較大的會議室。每次擴充時，新加入的空間必須為正方形且該邊長必須與相鄰的擴充前會議室邊長相同，如此才能確保合併後的會議室一定是四方形。以下圖為例，第一次擴充租用空間時，右邊編號為 $(1, 0)$ 的會議室空間會被跟編號為 $(0, 0)$ 的會議室合併。第二次擴充時，在 $(0, 0), (1, 0)$ 上方的四個 (2×2) 正方形小會議室會被合併進來。第三次擴充時，在 $(0, 0) \sim (0, 3)$ 左邊的 9 個 (3×3) 正方形小會議室會被合併進來。第四次擴充時，在 $(-3, 0) \sim (1, 0)$ 下方的 25 個 (5×5) 正方形小會議室會被合併進來。第五次擴充時，在 $(0, -5) \sim (0, 2)$ 右方的 64 個 (8×8) 正方形小會議室會被合併進來。後續的擴充則依此類推。



現在，若給定一個 n 的值，請計算第 n 次擴充時的正方形會議室的邊長。

輸入檔格式 (C:\room\input.txt)

輸入檔只有一個整數 n ， $n \leq 45$ 。

輸出格式 (C:\room\output.txt)

請輸出第 n 次擴充時的正方形會議室的邊長。

輸入檔範例 1

2

輸入檔範例 2

5

輸出範例 1

2

輸出範例 2

8

題目四：序列長度問題 (Sequence)

輸入檔：input.txt

輸出檔：output.txt

問題描述

傑倫與伊林非常喜歡上高中生物課，而生物老師孔智慧有一個「序列長度」問題，希望兩位同學幫忙計算。孔老師觀察出昆蟲體內存有許多過去從未被發現的奇特結構序列，這些結構序列是由一組元素組成，而且每一個元素在每一個結構序列中最多僅會出現一次。例如，當有 {A, B} 2 個不同組成元素時，則可以組成 A、B、A-B、B-A 等 4 種結構序列，這 4 種結構序列的長度總和為 6。而當有 {A, B, C} 3 個不同組成元素時，則可以組成 A、B、C、A-B、B-A、A-C、C-A、B-C、C-B、A-B-C、A-C-B、B-A-C、B-C-A、C-A-B、C-B-A 等 15 種結構序列，而這 15 種結構序列的長度總和為 33。由於預期陸續會有含新組成元素的結構序列被發現，孔老師因此希望傑倫與伊林幫忙計算看看當不同組成元素有 n 個時，所有可能的不同結構序列長度總和為何？聰明的你(妳)請寫一個程式幫傑倫與伊林來回答這個問題。

輸入檔格式 (C:\sequence\input.txt)

輸入一行只有一個正整數 n ($1 \leq n \leq 1000$)，代表不同組成元素。

輸出格式 (C:\sequence\output.txt)

輸出一個整數，即所有可能結構序列的總長度。

註：第二組輸出範例為一個 122 位的正整數，在本範例中因列印問題已自動更改格式為四行輸出，但在您的輸出檔中，仍應輸出至同一行中。

輸入檔範例 1

5

輸出範例 1

1305

輸入檔範例 2

80

輸出範例 2

153691304103615863187999023579932610872
824874638030049672444548946165482992214
601682328139903912967663080782140497997
96880

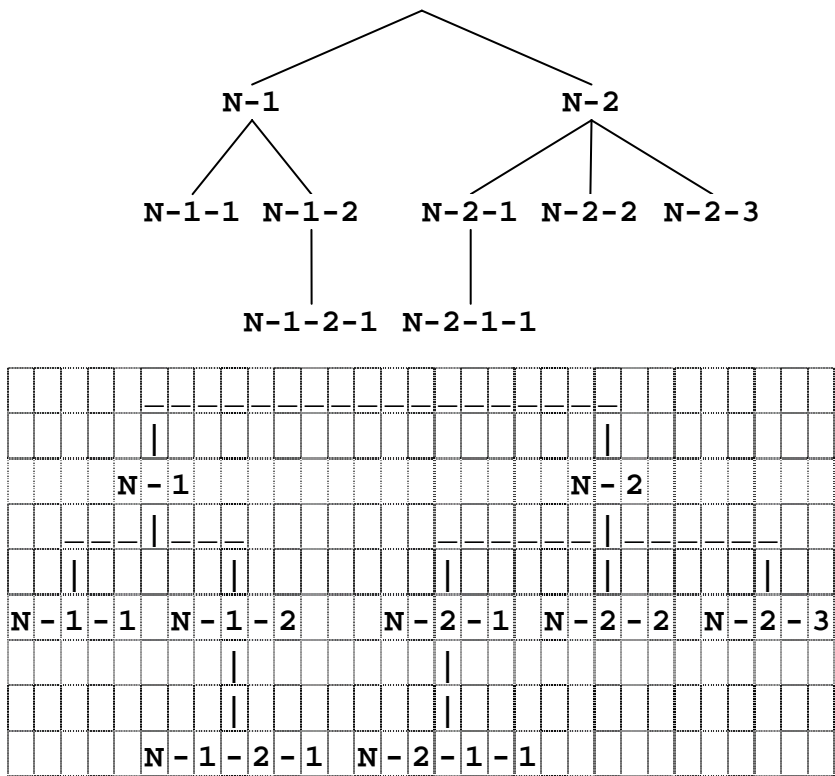
題目五：樹狀結構展示 (Tree)

輸入檔：input.txt

輸出檔：output.txt

問題描述

樹狀結構為一種程式設計者常用的資料結構，但較不容易在畫面上展示。常見的樹狀結構展示方式為「樹根在左、樹枝朝右」的橫向目錄型式，但其實若能以「樹根在上、樹枝朝下」的直向表示法，更能展現樹狀結構的精神。而要在電腦的二維螢幕上畫出美觀的樹狀結構，必須計算其中各個節點的座標位置。以下圖的樹狀結構為例，在一般文字模式的螢幕上（一個位置一個字元），其畫出來的樣子如下所示。其中名稱為 N-1 的節點，其起始座標為 (5,3)，名稱為 N-2 的節點，其起始座標為 (22,3)，名稱為 N-1-1 的節點，其起始座標為 (1,6)，其餘依此類推。



根據上例的表示方法，請寫一程式讀入如下格式的樹狀結構後，轉換為直向的樹狀結構，輸出每個節點的二維座標位置。

在將輸入轉換成直向的樹狀圖時，為求美觀，轉換時必須符合下列規則：

1. 節點以其名稱輸出。所有的距離、位置、寬度，都以「字元數」計算。
2. 同一階層的節點有相同垂直位置。
3. 同一階層的節點間必須求出最小間隔，但至少距離一個空白字元。
4. 若某一節點有多個子節點，則其水平位置必需在所有子節點的總寬度的正中央。其計算公式為： $S = \frac{1}{2}(L_S + L_E - L)$ 。

$$S = \frac{1}{2}(L_S + L_E - L)$$

其中：

S：本身節點的起始水平位置；當 S 無法整除時，取其四捨五入值。

- L_S ：第一個子節點的起始水平位置。
- L_E ：最後一個子節點的最後水平位置。
- L ：本身子節點的長度。

輸入檔格式 (C:\tree\input.txt)

1. 輸入為多行字串，每一行代表一個由根節點到本身節點的路徑，節點名稱由英文、數字與可列印符號表示，而上、下節點之間以冒號隔開。
2. 所有的輸入依照深度優先演算法(Depth-First)列出，不會有如下情形：
 - N-1**
 - N-1:N-1-1**
 - N-2**
 - N-1:N-1-2**
3. 輸入格式一定正確，不需考慮輸入格式錯誤情形。

輸出格式 (C:\tree\output.txt)

1. 輸出時，先輸出每個節點的路徑，其順序輸入相同。
2. 每個路徑後面，空一格後加上括號，括號內第一個值為該節點的起始水平座標位置，第二個值為其垂直座標位置。
3. 水平與垂直座標位置均以字元數計算，由 1 開始。如上例中的節點 N-1，其起始水平座標位置為 5，則輸出 5，垂直座標為 3，則輸出 3。

輸入檔範例 1

```
N-1
N-1:N-1-1
N-1:N-1-2
N-1:N-1-2:N-1-2-1
N-2
N-2:N-2-1
N-2:N-2-1:N-2-1-1
N-2:N-2-2
N-2:N-2-3
```

輸出範例 1

```
N-1 (5,3)
N-1:N-1-1 (1,6)
N-1:N-1-2 (7,6)
N-1:N-1-2:N-1-2-1 (6,9)
N-2 (22,3)
N-2:N-2-1 (15,6)
N-2:N-2-1:N-2-1-1 (14,9)
N-2:N-2-2 (21,6)
N-2:N-2-3 (27,6)
```

輸入檔範例 2

```
A
A:AB
A:AC
a
a:ab
a:ac
```

輸出範例 2

```
A (3,3)
A:AB (1,6)
A:AC (4,6)
a (9,3)
a:ab (7,6)
a:ac (10,6)
```