

模組 F – 演算法模組

注意事項：為與國際競賽接軌，本次競賽不提供本地端 XAMPP 環境，選手需將成果傳送至遠端評分系統，確認作答成果。

近年來，愈來愈多商業系統採用網頁應用程式的方式呈現，在不同的行業中，都有該行業需要解決的問題，許多的問題背後都有不同的演算法，在這個模組要測試你是否有可以解決這些問題。

本模組採即時自動評分。實際競賽試題與測試資料會有 30% 的變動。

工作要求

你必須從 STDIN 讀取各項的測試資料，並輸出至 STDOUT，每個題組有多個測試資料，必須全部通過才有分數，輸出中不可有多餘的文字。

各題的執行時間必須在 10 秒內完成，使用的記憶體大小必須少於 128MB。

範例題目

題目

輸入有多行，每行有兩個正整數 a b ，中間使用單一空白分隔，請輸出 $a+b$ 的結果，若 a 為 -1 時表示測資輸入結束。

測試資料

輸入	輸出
1 2 3 4 -1	3 7
5 9 0 9 0 0 5 1000 -1	14 9 0 1005

範例程式碼

```
<?php
while(($line = readline()) !== false) {
    $numbers = explode(' ', $line);
    $num1 = $numbers[0];

    if($num1 === '-1')
        break;

    $num2 = $numbers[1];

    echo $num1 + $num2;
    echo "\n";
}
```

題目 1

題目

計算最大公因數

給定一串數字，計算其最大公因數 (greatest common divisor)。

測試資料第一行會有一個正整數 n ，接下來有 n 行文字，每行有一個數字 i ， $1 \leq i \leq 2^{31}-1$ ，請輸出 n 個數字之間的最大公因數。

測試資料

輸入	輸出
3 50 75 100	25
4 29 11 19 50	1

題目 2

題目

找出眾數

給定一串數字，找出這串數字的眾數。眾數為一組資料中出現次數最多的數值。眾數可能是一個數，但也可能是多個數。若資料的數值出現次數相同且無其他數值時，則不存在眾數。

測試資料第一行會有一個正整數 n ，接下來有 n 行文字，每行有一個數字 i ，

$1 \leq i \leq 2^{31}-1$ ，請輸出 n 個數字之間的眾數，當有多組時，請由數值小到大每行輸出一個，若不存在，則輸出-1。

測試資料

輸入	輸出
3 50 75 100	-1
5 5 5 2 8 2	2 5

題目 3

題目

判斷完美數

當一個數字的所有的真因子（即除了自身以外的因數）的和剛好為本身時，即為完美數，請判斷輸入的數字是否為完美數。

測試資料第一行會有一個正整數 n ，接下來有 n 行文字，每行有一個數字 i ，

$1 \leq i \leq 2^{63}-1$ ，若 i 為完美數則輸出 Y，否則輸出 N。

測試資料

輸入	輸出
3	N
4	Y
28	N
100	

題目 4

題目

匯率換算

給定銀行的牌告匯率，進行各種貨幣與新台幣的轉換。

測試資料第一行會有一個正整數 m ，接下來有 m 行文字，每行有一個數字 $o a b c d$ ， o 為三位數英文貨幣代碼， a 、 b 、 c 、 d 精準到小數點第五位，其中 a 與 b 為 現金匯率， c 與 d 為即期匯率， a 與 c 為買入匯率， b 與 d 為賣出匯率。

接下來有一個數字 n ，表示接下來有 n 個客戶，每行有四個部分 $w x y z$ 。

- w 為交易方式， A 代表使用銀行帳戶交易， C 代表使用現金交易。
- x 為兌換前貨幣別
- y 為兌換後貨幣別
- z 為兌換數量， $z \geq 0$

x 、 y 至少一個為 TWD。

對於每個客戶，請輸出其交易後的貨幣數量，精準到小數點第五位，無條件捨去，不足五位補 0。

測試資料

輸入	輸出
2 TWD 1 1 1 1 USD 27 28 27.5 27.711 2 C USD TWD 100 A TWD USD 2700	2700.00000 97.43423
2 TWD 1 1 1 1 MYR 5.5 7.25 0 0 2 C TWD MYR 10 C MYR TWD 55	55.00000 7.58620

題目 5

題目

Luhn 演算法 (Luhn algorithm)，是一種簡單的校驗和演算法，一般用於驗證身份識別碼，例如發卡行識別碼、國際行動裝置辨識碼 (IMEI)，美國國家提供商標識號碼，或是加拿大社會保險號碼。你現在要判斷輸入的數字是否可以通過檢核。

Luhn 演算法會通過校驗碼對一串數字進行驗證，校驗碼會被加到這串數字的末尾處，從而得到一個完整的身份識別碼。

以數字「79927398713」為例，為計算其校驗位是否正確，我們先取前 N-1 個數字，也就是「7992739871」進行計算：

1. 從從右往左，奇數位乘 2 (例如， $1*2=2$)，然後將兩位數字的個位與十位相加 (例如，16： $1+6=7$ ，18： $1+8=9$)
2. 把得到的數字加在一起 (本例中得到 67)
3. 將該數字乘以 9 (603)
4. 取其個位數字 (3)，得到校驗位
5. 將計算出來的校驗位與輸入的校驗位進行比較判斷是否一致

原始數字	7	9	9	2	7	3	9	8	7	1	
乘 2	7	18	9	4	7	6	9	16	7	2	
將數字相加	7	9	9	4	7	6	9	7	7	2	=67

測試資料第一行會有一個正整數 n，接下來有 n 行文字，每行有一組要檢查的數字，每行會有任意長度的文字，只有數字或空格才是正常的輸入內容，若包含不合法的字元，或是無法通過 Luhn 的檢查，請輸出 N，若通過檢查，則輸出 Y。

測試資料

輸入	輸出
3 79927398713 79927398714 1234567831	Y N N
4 4291 1966 1372 1345 1966 1372 123AB0	Y Y N N

題目 6

題目

密碼規則檢查

強密碼可以降低密碼被猜中的機率，假設密碼可以由不同類型的字元組成：

- 數字 0123456789
- 大寫英文 ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 小寫英文 abcdefghijklmnopqrstuvwxyz
- 特殊符號 ~!@#\$%^&*()_+ = - \ | ' ; : / . , ? > <

請依照以下規則判斷密碼強度：

- 四種類型的組合都有: 4
- 四種組合中只有三個: 3
- 四種組合中只有兩個: 2
- 四種組合中只有一個: 1

測試資料第一行會有一個正整數 n ，接下來有 n 行文字，每行有一串文字 s ，每行請輸出該密碼的強度。

測試資料

輸入	輸出
4	3
45Af	4
Qwe!@#123	1
100	2
QaQ	

題目 7

題目

提供給你一組數字的順序，試著排列他們，找到下一個大的數值。若沒有的話，則回傳最小的數值。

測試資料第一行會有一個正整數 m ， $1 \leq m \leq 10$ ，代表接下來有 m 組資料。每組資料有兩行，第一行有一個正整數 n ， $1 \leq n \leq 100$ ，代表接下來有 n 個數字，下一行會有 n 個數字 $num[i]$ ， $1 \leq i \leq n$ ， $0 \leq num[i] \leq 9$ ，數字之間用空白分隔。

每組資料輸出一行結果，每行輸出一個數值。

測試資料

輸入	輸出
2 3 2 1 3 3 3 2 1	231 123
2 4 1 5 1 1 1 9	5111 9

題目 8

題目

Hash table 在撰寫一些邏輯時時常使用，但遇到非數字的 key 時需要做一些轉換才可以用成數值表達出他的 hash value，在這裡你需要依以下規則實作字串的 hash function。

這個 hash function 已經於其他語言實做出來了，由於該語言使用二進位儲存數值，因此有數值表達範圍的限制 ($-2^{31} \sim 2^{31}-1$)，在你的實作中，也必須模擬出這個行為，以讓計算出來的 hash value 可以於不同語言中使用。

給定一字串 str，長度為 n，str[i] 表示該字串中第 i 個字元的 ASCII code，hash value 的計算方式為：

$$\text{str}[0]*31^{(n-1)} + \text{str}[1]*31^{(n-2)} + \dots + \text{str}[n-1]*31^{(0)}$$

輸入的第一行有一個正整數 n，接下來有 n 行文字，每行不超過 65536 個字元，請輸出該字串的 hash value。

測試資料

輸入	輸出
5 a abcd abcde abcdef abcdefg	97 2987074 92599395 -1424385949 -1206291356
1 Aaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaa (備註: 100 個 a)	-323643840

題目 9

題目

給定一個只包含 '('、')' 和 '*' 三種類型字元的字串，寫一個程式來檢查這個字符串是否有效。一個合法的字串必須符合以下規則：

1. 任何左括號 '(' 必須有一個相應的右括號 ')'。
2. 任何右括號 ')' 必須有一個相應的左括號 '('。
3. 左小括號 '(' 必須放在相應的右小括號 ')' 之前。
4. '*' 可以被視為單個右括號 ')' 或單個左括號 '(' 或空白。
5. 空白也是有效的。

輸入的第一行有一個正整數 n ，接下來有 n 行文字，每行為一個要驗證的字串，字串長範圍為 $[1, 100]$ ，對於合法的字串，請輸出 Y，否則請輸出 N

測試資料

輸入	輸出
5	Y
()	Y
(*)	Y
((*)	Y
((*)*	N
((*))*	

題目 10

題目

計算最佳的股票利益。

給定一個整數序列 $p[i]$ ，長度為 n ，裡面包含第 i 天的股價，由於資金有限，買入後必須要賣出，才可以進行下一次買賣，也不可以賣出沒有擁有的股票。

請計算出經過最多 m 次的買與賣後，可以獲得最多的利益為多少。

輸入的第一行有兩個正整數 m n ， m 代表最多可以買與賣幾次， n 代表下一行有 n 個數字，每個數字代表第 i 天的股價，請你輸出最多進行 m 次買賣後，可以獲得的最高利益。

測試資料

輸入	輸出
2 4 9 4 8 7	4 (第二天買第三天賣) $\cdot (8-4) = 4$
2 4 9 4 8 9	5 (第二天買第三天賣，接著同天買入後，第四天賣出) $\cdot (8-4) + (9-8) = 5$
1 4 9 4 8 9	5 (第二天買第四天賣) $\cdot (9-4) = 5$

選手注意事項

以下說明時用到 XX 代表選手個人的崗位編號，Y 代表模組編號

- 將完成的結果上傳至評分網站，評分網站網址為 <http://judge.web>，並且在競賽時間結束前評分完成
- 將完成的結果**備份**至網站根目錄，用 XX_Module_Y 作為資料夾名稱，檔名為 pZZ.php，ZZ 為題目編號，因網頁無 STDIN，因此使用瀏覽器瀏覽網頁無法執行是正常情況
- 各題的執行時間必須在 10 秒內完成，使用的記憶體大小必須少於 128MB。