



# 全國高級中等學校 110 學年度商業類學生技藝競賽

## 【程式設計】職種 【術科】正式試題卷

崗位編號：\_\_\_\_\_ 姓名：\_\_\_\_\_

- A 二數有權重的相加
- B 閏年 (多列版)
- C 所有位數加減加減
- D 最大偶數和
- E 字典
- F 抱怨值問題
- G 猜數字
- H 校驗和
- I 123
- J 噁爛數
- K 求餘數
- L 質數
- M 矩陣的直積
- N 字串編輯距離
- O 最長共同子序列 LCS(Longest common subsequence)
- P 錢幣
- Q 果子堆合併
- R 條條大路通羅馬

# 【程式設計】 職種 【術科】 正式試題

在英文裡面，row是橫的，column是直的

列是row，行是column

在繁體中文版的Excel裡面，row叫做「列」，column叫做「欄」。

## A 二數有權重的相加

輸入二個數字 $a$   $b$  有權重的相加，輸出:  $4 * a + 6 * b$ 。

### 輸入

輸入有若干列，每列為一組測試資料。輸入的每一列有二個整數。以空白隔開的正整數。輸入以 EOF 作為結束。

### 輸出

對於所輸入的每一列有二個整數，要各別輸出一列，依公式有權重的相加。

### 範例輸入輸出

範例輸入

1	1	1
2	10	100

範例輸出

1	10
2	640

## B 閏年 (多列版)

西元年份除以4不可整除，為平年。False

西元年份除以4可整除，且除以100不可整除，為閏年。True

西元年份除以100可整除，且除以400不可整除，為平年。False

西元年份除以400可整除，為閏年。True

以下為舉例說明：

除了不是400的倍數的100的倍數以外，四的倍數的年份均為閏年，閏年這年會多一天 (2月29日)。

任何能以 4 整除的年份都是閏年：例如 1980 年、1984 年、1988 年、1992 年及 1996 年都是閏年。

不過，仍必須將一個小錯誤列入考量。西曆規定能以 100 (例如1900 年) 整除的年份，同時也要能以 400 整除，才算是閏年。

下列年份不是閏年：1700、1800、1900、2100、2200、2300、2500、2600。

原因是這些年份能以 100 整除，但無法以 400 整除。

下列年份為閏年：1600、2000、2400。

閏年，請輸出「a leap year」，否則請輸出「a normal year」

### Input

輸入的第一列有一個整數  $n$ 。接下來的  $n$  列每列有一個正整數  $y$ ，代表西元年份。

```
1 | 4
2 | 1992
3 | 1991
4 | 1700
5 | 2400
```

### Output

對於所輸入的每個  $y$ ，要各別輸出一列。若  $y$  是閏年，請於該行輸出「a leap year」，否則請輸出「a normal year」。

```
1 | a leap year
2 | a normal year
3 | a normal year
4 | a leap year
```

## C 所有位數加減加減

給你一個正整數，輸出所有位數加減加減的總和，例如12345，為5位數  
 $1 + 2 - 3 + 4 - 5 = -1$ ，則輸出 $-1$ 。1997，為4位數 $1 + 9 - 9 + 7 = 8$ ，則輸出8。

### 輸入

輸入有若干列。輸入的每一列有一個正整數，每一列為一筆的測試資料，內容為一個正整數 $N$ ， $1 \leq N \leq 2^{31} - 1$ 。輸入以EOF作為結束。

### 輸出

請輸出  $N$  所有位數加減加減的總和。

### 範例輸入輸出

範例輸入

1	12345
2	1997
3	1

範例輸出

1	-1
2	8
3	1

## D 最大偶數和

給 $n$ 個整數，在這些數字中，求其最大偶數和。如 $n$ 個整數總和是偶數，直接輸出。如果總和是奇數，那麼 $n$ 個整數中至少有一個奇數；然後要知道從這 $n$ 個整數中剔除偶數是沒有意義的；所以做法就是從總和中減去一個奇數，在其他的 $n - 1$ 個整數，求其最大偶數和。

### 輸入

輸入的第一列包含一個整數  $n(1 \leq n \leq 100000)$ 。下一列包含 $n$ 個整數，以空白隔開的正整數。這些正整數中的每一個都在1 到  $10^9$  的範圍內。

### 輸出

給定 $n$ 個的整數，在這些數字中，求其最大偶數和。

### 範例輸入輸出

範例輸入 I

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

範例輸出 I

```
1 | 6
```

範例輸入 II

```
1 | 5
2 | 99999999 99999999 99999999 99999999 99999999
```

範例輸出 II

```
1 | 399999996
```

## E 字典

字典，寫一個程式將外國文字翻譯成英文字。

### 輸入

輸入首先是字典的內容，每一列會有兩個字串(單字)，最多不會超過100000列。每列包含2個字串(單字)（全部小寫且長度不會超過10個字元），第1個是英文字，第2個是外國文字，會先給你每個英文字對應到的外國文字，對應的字保證是一對一，形成一個字典。在字典中，沒有任一個外國字串單字會出現超過1次。

接著有一空白列，然後才是需要翻譯的外國文字（每個一列），接下來會有多筆詢問，輸入一個外國文字，輸出相對應的英文字，如果不存在則輸出 eh。輸入以 EOF 作為結束。

### 輸出

將外國文字翻譯成英文字輸出，如果字典找不到則輸出 eh

### 範例輸入輸出

範例輸入 I

```
1 dog ogday
2 cat atcay
3 pig igpay
4 froot ootfray
5 loops oopslay
6
7 atcay
8 ittenkay
9 oopslay
```

範例輸出 I

```
1 cat
2 eh
3 loops
```

範例輸入 II

```
1 dog ogday
2 cat latcay
3 pig igpay
4 froot ootfray
5 loops oopslay
6 kitten abooskay
7 paper koopsmay
8
9 latcay
10 ittenkay
11 oopslay
12 abooskay
13 kitten
14 booskay
15 koopsmay
```

## 範例輸出 II

```
1 cat
2 eh
3 loops
4 kitten
5 eh
6 eh
7 paper
```

## F 抱怨值問題

### 抱怨值問題

給予一串整數來表示排隊買票入場號碼牌的優先順序。如果拿到1號號碼牌的人就表示可以第一個買票入場。在排隊中，抱怨值是用排隊買票號碼牌來計算，假如有人拿的號碼牌比你後面（號碼牌較大）卻比你先買票入場的話，抱怨值就是 +1。

```
1 20 54 1 50 100 2000 50
  ↑  ↑  ↑
```

1號號碼牌，前面號碼牌較大的有20 54，抱怨值為2。

```
1 20 54 1 50 100 2000 50
  ↑      ↑
```

50號號碼牌，前面號碼牌較大的有54，抱怨值為1。

```
1 20 54 1 50 100 2000 50
  ↑      ↑  ↑  ↑
```

50號號碼牌，前面號碼牌較大的有54 100 2000，抱怨值為3。  
這筆的測試資料抱怨值合計2+1+3為6。

## 輸入

測資第一列給定一正整數  $n$  ( $2 \leq n \leq 10^4$ )，代表有  $n$  個人排隊買票。第二列給定  $n$  個正整數  $a$  ( $1 \leq a \leq 65535$ )，為每筆的測試資料，以空白隔開的正整數，因作業錯誤關係，會有重複的號碼牌且沒有連號。每個輸入的測試檔案，只有一筆測試資料。

## 輸出

每筆測試資料輸出一列。輸出抱怨值。

## 範例輸入輸出

範例輸入 I



```
1 | 8
2 | 1 20 54 1 50 100 2000 50
```

範例輸出 I

```
1 | 6
```

範例輸入 II

```
1 | 7
2 | 25 100 50 3 122 20 1001
```

範例輸出 II

```
1 | 8
```

## G 猜數字

電腦設定猜測數字一組四碼的數字作為謎底，另一方為猜測數字。每猜一個猜測數字，電腦就要根據猜測數字和這個猜測數字給出提示，提示以 $xAyB$ 形式呈現，其中 $x$ 表示位置正確的數的個數，而 $y$ 表示數字正確而位置不對的數的個數。

例如，當電腦設定正確答案為3**1**28，而另一方猜145**2**時，電腦必須提示0A2B。

例如，當電腦設定正確答案為56**3**4，而另一方猜87**3**1時，電腦必須提示1A0B。

若是允許重複數字存在的猜數字，其規則較複雜。其規則如下：

除了上面的規則外，如果有出現重複的數字，則重複的數字每個也只能算一次，且以最優的結果為準，如電腦設定正確答案為5567，另一方猜測數字為5455，則在這裡不能認為猜測數字的第一個5對正確答案第二個，根據最優結果為準的原理和每個數字只能有一次的規則，兩個比較後，應該為1A1B，第一個5位置正確，記為1A；猜測數字中的第三個5或第四個5和正確答案的第二個5匹配，只能記為1B。當然，如果有猜測數字5234中的第一個5不能與正確答案中的第二個5匹配，因此只能記作1A0B。

## 輸入

多筆輸入。第一列有四個介於0-9之間的數字，代表電腦設定正確答案。第二列有一個整數 $n$ ， $1 \leq n \leq 20$ ，代表接下來有 $n$ 列猜測數字，每列有四個介於0-9之間的數字，每列各代表一組嘗試猜測數字。

## 輸出

輸出 $n$ 列。對於每組嘗試猜測數字，若有 $x$ 個數字的值正確，且在正確的位子上，另外有 $y$ 個數字的值正確，但不在正確的位子上，輸出 $xAyB$ 。

正確答案：1101 猜測數字：1010，這樣要算1A2B，所以每次再檢查B次數的時候，要把A排除。

## 範例輸入輸出

範例輸入 I

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

## 範例輸出 I

1	1A1B
2	2A2B
3	2A0B
4	0A4B

## 範例輸入 II

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

## 範例輸出 II

1	3A0B
2	0A0B
3	1A1B
4	2A2B

## H 校驗和

Checksum(校驗和)是為了保證 Header(標頭) 或 Data(資料) 不被破壞而發展出來的機制，IP層有IP層的Header Checksum(標頭校驗和)，TCP層有TCP層的Checksum(校驗和)。給定IP層Header(標頭)的20Bytes的資料，計算IPv4 Header Checksum(標頭校驗和)。

把資料以2Bytes為一組，20Bytes的資料做加總：

20Bytes的資料，要計算Header Checksum(標頭校驗和)，目前Header Checksum先代入00 00 45 00 00 30 cc 61 40 00 40 06 **00 00** 0a 05 04 6b 0a 08 09 ed

- 做加總  
 $4500 + 0030 + cc61 + 4000 + 4006 + \mathbf{0000}$ (要計算Header Checksum) +  $0a05 + 046b + 0a08 + 09ed = 1b3fc$
- 以2Bytes為一組，進位的再加回來  
 $1 + b3fc = b3fd$  (1011 0011 1111 1101)
- b3fd的結果取1的補數  
 $0100\ 1100\ 0000\ 0010 \rightarrow 4c\ 02$  (計算出Header Checksum)

20 Bytes的資料，b861為Header Checksum(標頭校驗和)

45 00 00 73 00 00 40 00 40 11 **b8 61** c0 a8 00 01 c0 a8 00 c7

- 做加總  
 $4500 + 0073 + 0000 + 4000 + 4011 + \mathbf{b861} + c0a8 + 0001 + c0a8 + 00c7 = 2fffd$
- 以2Bytes為一組，進位的再加回來  
 $2 + fffd = ffff$
- ffff的結果取1的補數  
取1的補數（翻轉每1位元）得到0000，這表示IP層的Header Checksum(標頭校驗和)沒有檢測到錯誤。

## 輸入

第一列的數字 $n$ 代表有幾組資料要測試， $1 \leq n \leq 100$ ，第二列起為每組的測試資料，接下來的 $n$ 列，給定IP層Header(標頭)的20Bytes的資料，計算IPv4 Header Checksum(標頭校驗和)。

## 輸出

對於所輸入的每一列，要各別輸出一列，輸出Header Checksum(標頭校驗和)，十六進制使用小寫英文字母。

## 範例輸入輸出

## 範例輸入 I

```
1 5
2 45 00 00 30 cc 61 40 00 40 06 00 00 0a 05 04 6b 0a 08 09 ed
3 45 00 00 73 00 00 40 00 40 11 b8 61 c0 a8 00 01 c0 a8 00 c7
4 45 00 00 47 73 88 40 00 40 06 00 00 83 9f 0e 85 83 9f 0e a1
5 45 00 00 47 73 88 40 00 40 06 a2 c4 83 9f 0e 85 83 9f 0e a1
6 45 00 05 14 42 a2 21 40 80 01 00 00 c0 a8 00 03 c0 a8 00 01
```

## 範例輸出 I

```
1 4c02
2 0000
3 a2c4
4 0000
5 50b2
```

# I 123

小朋友剛學會寫英文的"one two three"。小朋友在紙上寫了很多這幾個字，而你的工作便是辨認它們。要注意的是小朋友是個小孩子，因此他會犯些小錯誤：至多一個錯誤的字母拼錯，但是單字長度永遠正確。小朋友所寫的一定是小寫字母，每個單字只會有一種對"one two three"合理解釋。

## 輸入

第一行為小朋友所寫單字數。接下來的每一行為英文的"one two three"小寫字母組成的單字。單字必符合上述限制：至多一個錯誤的字母拼錯，但是單字長度永遠正確。

## 輸出

對每筆測資，輸出英文單字"one two three"對應到的數字1 2 3。

## 範例輸入輸出

### 範例輸入 I

1	3
2	owe
3	too
4	theee

### 範例輸出 I

1	1
2	2
3	3

### 範例輸入 II

1	9
2	owe
3	too
4	theee
5	tne
6	owo
7	thfee
8	one
9	two
10	three

## 範例輸出 II

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

## J 噁爛數

什麼是噁爛數？

如果一個數的質因數沒有 2, 3, 5 以外的數, 它就是噁爛數.

例如:

1024 的質因數只有 2, 沒有(2, 3, 5)以外的數, 它是噁爛數.

235 的質因數有 5 和 47, 有(2, 3, 5)以外的數, 它不是噁爛數.

450 的質因數有 2, 3, 5 沒有(2, 3, 5)以外的數, 它是噁爛數.

245 的質因數有 5 和 7, 有(2, 3, 5)以外的數, 它不是噁爛數.

## 輸入

測資第一列給定一正整數  $n$  ( $1 \leq n \leq 10^6$ ) , 代表有  $n$  個正整數。第二列給定  $n$  個正整數  $a$  ( $1 \leq a \leq 10^6$ ) , 以空白隔開的正整數, 為每筆的測試資料。每個輸入的測試檔案, 只有一組測試資料。

## 輸出

輸出 $n$ 行, 正整數  $a$  ( $1 \leq a \leq 10^9$ ) , 正整數  $a$ 為噁爛數輸出True, 否則回傳 False。

## 範例輸入輸出

範例輸入 I

```
1 | 4
2 | 1024 235 450 245
```

範例輸出 I

```
1 | True
2 | False
3 | True
4 | False
```



## K 求餘數

計算  $R = X^Y \bmod M$ ，

用VB來求餘數時，可以用mod這個關鍵字來實作。但如果給  $X, Y, M$ ，要算出餘數  $R$ ，當  $X$  或  $Y$  很大時，求餘數就變得不簡單了。現在，請設計一個程式，來解決求餘數的問題。

首先，要知道數學式子： $(A*B)\%C = (A\%C) * (B\%C)$ 。因此我就可以不用把  $B^P$  算完再去對  $M$  取餘數(避免超過變數範圍)，但是如果是  $((((B\%M)*B)\%M)*B)\%M$ .....這樣乘的話會TLE的，所以算次方請用次方除二相乘的遞迴來算次方，也就是  $(B^P)\%M = (B^{(P/2)}\%M) * (B^{(P/2)}\%M)$  這樣遞迴。

計算  $B^P$

$P$  為偶數，答案為  $(B^{(P/2)})^2$

$P$  為奇數，答案為  $B * (B^{(P//2)})^2$

## 輸入

每組測試資料有3列，每一列包含一個數字為整數，分別為  $X, Y, M$ ，

$0 \leq X, Y \leq 2147483647, 1 \leq M \leq 46340$ ，例如：

10

2009

9

代表  $X = 10, Y = 2009, M = 9$ 。每組測試資料以一行換行作為區隔。輸入以 EOF 作為結束。

## 輸出

每筆測試資料輸出一列。對每一筆測試資料輸出餘數  $R = X^Y \bmod M$ 。

## 範例輸入輸出

範例輸入 I

1	10
2	2009
3	9
4	
5	2
6	99
7	5
8	
9	
10	3
11	18132
12	17
13	
14	17
15	1765
16	3

範例輸出 I

1	1
2	3
3	13
4	2

範例輸入 II

1	65535
2	65535
3	36123
4	
5	2374859
6	3029382
7	36123

範例輸出 II

1	14211
2	13195

## L 質數

質數的定義為：除了1和它本身之外，沒有別的數可以整除它的。（請注意：在本問題中，1被定義為質數）

你的任務是，給你 $N$ 及 $C$ ，請你找出1到 $N$ 中所有的質數，並把他們排成一列（假設共有 $K$ 個）。如果 $K$ 是偶數，請輸出中間那 $C * 2$ 個質數。如果 $K$ 是奇數，則輸出中間那 $(C * 2) - 1$ 個質數。

### 輸入

每組測試資料一行，各含有2個整數 $N, C$ 。（ $1 \leq N \leq 1000, 1 \leq C \leq N$ ）。輸入以 EOF 作為結束。

### 輸出

對每組測試資料輸出 $N \ C$ ：，然後輸出題目要求的質數。每個數前方有一空格。如果 $2 * C$ 或 $(2 * C) - 1$ 大於等於 $K$ ，就把他們全部列出（如第3個sample）。每組測試資料後亦請空一行。請參考Sample Output。

### 範例輸入輸出

範例輸入 I

```
1 | 21 2
2 | 18 2
3 | 18 18
4 | 100 7
```

範例輸出 I

```
1 | 21 2: 5 7 11
2 |
3 | 18 2: 3 5 7 11
4 |
5 | 18 18: 1 2 3 5 7 11 13 17
6 |
7 | 100 7: 13 17 19 23 29 31 37 41 43 47 53 59 61 67
```

## M 矩陣的直積

給定任兩個矩陣  $A$  和  $B$ ，我們可以得到兩個矩陣的直積，或稱為克羅內克乘積  $A \otimes B$ ，其定義如下

$$\begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}$$

當  $A$  是一  $m \times n$  矩陣和  $B$  是一  $p \times r$  矩陣時， $A \otimes B$  會是一  $mp \times nr$  矩陣，而且此一乘積是不可交換的。

舉些例子：

$$\begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 3 & 2 \cdot 0 & 2 \cdot 3 \\ 1 \cdot 2 & 1 \cdot 1 & 2 \cdot 2 & 2 \cdot 1 \\ 3 \cdot 0 & 3 \cdot 3 & 1 \cdot 0 & 1 \cdot 3 \\ 3 \cdot 2 & 3 \cdot 1 & 1 \cdot 2 & 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 0 & 6 \\ 2 & 1 & 4 & 2 \\ 0 & 9 & 0 & 3 \\ 6 & 3 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 & 2 \\ 6 & 3 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 1 \cdot 2 & 2 \cdot 0 & 2 \cdot 5 & 2 \cdot 2 \\ 1 \cdot 6 & 1 \cdot 3 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 3 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 3 \cdot 2 & 4 \cdot 0 & 4 \cdot 5 & 4 \cdot 2 \\ 3 \cdot 6 & 3 \cdot 3 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 3 & 4 \cdot 7 \\ 1 \cdot 0 & 1 \cdot 5 & 1 \cdot 2 & 0 \cdot 0 & 0 \cdot 5 & 0 \cdot 2 \\ 1 \cdot 6 & 1 \cdot 3 & 1 \cdot 7 & 0 \cdot 6 & 0 \cdot 3 & 0 \cdot 7 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 5 & 2 & 0 & 10 & 4 \\ 6 & 3 & 7 & 12 & 6 & 14 \\ 0 & 15 & 6 & 0 & 20 & 8 \\ 18 & 9 & 21 & 24 & 12 & 28 \\ 0 & 5 & 2 & 0 & 0 & 0 \\ 6 & 3 & 7 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \times 0 & 1 \times 5 & 2 \times 0 & 2 \times 5 \\ 1 \times 6 & 1 \times 7 & 2 \times 6 & 2 \times 7 \\ 3 \times 0 & 3 \times 5 & 4 \times 0 & 4 \times 5 \\ 3 \times 6 & 3 \times 7 & 4 \times 6 & 4 \times 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -4 & 7 \\ -2 & 3 & 3 \end{bmatrix} \otimes \begin{bmatrix} 8 & -9 & -6 & 5 \\ 1 & -3 & -4 & 7 \\ 2 & 8 & -8 & -3 \\ 1 & 2 & -5 & -1 \end{bmatrix} = \begin{bmatrix} 8 & -9 & -6 & 5 & -32 & 36 & 24 & -20 & 56 & -63 & -42 & 35 \\ 1 & -3 & -4 & 7 & -4 & 12 & 16 & -28 & 7 & -21 & -28 & 49 \\ 2 & 8 & -8 & -3 & -8 & -32 & 32 & 12 & 14 & 56 & -56 & -21 \\ 1 & 2 & -5 & -1 & -4 & -8 & 20 & 4 & 7 & 14 & -35 & -7 \\ -16 & 18 & 12 & -10 & 24 & -27 & -18 & 15 & 24 & -27 & -18 & 15 \\ -2 & 6 & 8 & -14 & 3 & -9 & -12 & 21 & 3 & -9 & -12 & 21 \\ -4 & -16 & 16 & 6 & 6 & 24 & -24 & -9 & 6 & 24 & -24 & -9 \\ -2 & -4 & 10 & 2 & 3 & 6 & -15 & -3 & 3 & 6 & -15 & -3 \end{bmatrix}$$

## 輸入

輸入整數不會超過  $\pm 65535$

兩矩陣大小不會超過  $20 \times 20$

每組測資第一行四個數字  $a \ b \ c \ d$

代表第一個矩陣有  $a$  列  $b$  行

第二個矩陣有  $c$  列  $d$  行

接下來  $a$  列，每列  $b$  個數字

$c$  列，每列  $d$  個數字

每個數字以空白隔開。每組測資有 2 個矩陣，請把他們兩個矩陣的直積，或稱為克羅內克乘積  $A \otimes B$  的結果輸出。

每個輸入的測試檔案，只有一筆測試資料。

## 輸出

當  $A$  是一  $m \times n$  矩陣和  $B$  是一  $p \times r$  矩陣時， $A \otimes B$  會是一  $mp \times nr$  矩陣，輸出矩陣，每個數字以空白隔開。

## 範例輸入輸出

範例輸入 I

```
1 | 2 2 2 2
2 | 1 2
3 | 3 1
4 | 0 3
5 | 2 1
```

範例輸出 I

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

### 範例輸入 II

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

### 範例輸出 II

1	0 5 2 0 10 4
2	6 3 7 12 6 14
3	0 15 6 0 20 8
4	18 9 21 24 12 28
5	0 5 2 0 0 0
6	6 3 7 0 0 0

### 範例輸入 III

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

### 範例輸出 II

0 5 0 10
6 7 12 14
0 15 0 20
18 21 24 28

### 範例輸入 IV

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

### 範例輸出 IV

1	8	-9	-6	5	-32	36	24	-20	56	-63	-42	35
2	1	-3	-4	7	-4	12	16	-28	7	-21	-28	49
3	2	8	-8	-3	-8	-32	32	12	14	56	-56	-21
4	1	2	-5	-1	-4	-8	20	4	7	14	-35	-7
5	-16	18	12	-10	24	-27	-18	15	24	-27	-18	15
6	-2	6	8	-14	3	-9	-12	21	3	-9	-12	21
7	-4	-16	16	6	6	24	-24	-9	6	24	-24	-9
8	-2	-4	10	2	3	6	-15	-3	3	6	-15	-3

## N 字串編輯距離

兩個字串之間的編輯距離是將一個字串轉換為另一個字串所需的最少操作次數。

字串轉換允許的操作有：

在字符串中添加一個字元。

從字符串中刪除一個字元。

替換字符串中的一個字元。

您的任務是計算兩個字符串之間的編輯距離。

舉些例子：

Input: "love", "movie"

Output: 2

love -> move (可以先把l換成m)

move -> moive (添加一個字元i)

Input: "horse", "ros"

Output: 3

horse -> rorse (可以先把h換成r)

rorse -> rose (刪除一個字元r)

rose -> ros (刪除一個字元e)

Input: "intention", "execution"

Output: 5

intention -> inention (刪除一個字元t)

inention -> enention (替換字符串中的一個字元 i -> e)

enention -> exention (替換字符串中的一個字元 n -> x)

exention -> exection (替換字符串中的一個字元 n -> c)

exection -> execution (添加一個字元u)

## 輸入

第一個輸入行有一個包含 a-z 之間的  $n$  個字元的字串。

第二個輸入行有一個包含 a-z 之間的  $m$  個字元的字串。

字串內容為小寫的英文字母。

## 輸出

輸出一個整數：字符串之間的編輯距離。

## 範例輸入輸出



### 範例輸入 I

```
1 | love
2 | movie
```

### 範例輸出 I

```
1 | 2
```

### 範例輸入 II

```
1 | horse
2 | ros
```

### 範例輸出 II

```
1 | 3
```

### 範例輸入 III

```
1 | intention
2 | execution
```

### 範例輸出 III

```
1 | 5
```

## O 最長共同子序列LCS(Longest common subsequence)

「最長共同子序列」 (Longest Common Subsequence, LCS) 。 LCS 是兩個序列(sequence) , 在各自所有的子序列( subsequence)之中, 一模一樣而且最長的那個子序列(subsequence)。

給2個字串, 請你輸出他們的最長共同子序列 (Longest common subsequence) 的長度。

也就是說, 在這兩個字串各自所有的子序列之中, 內容相同而且長度最長的那個子序列。舉

例來說有兩個字串abcdghxy 和aedfhrz, 它們的最長共同子序列為adh, 長度為3。

一般LCS問題( Longest Common subsequence, 最長共同子字串)就是給定兩個字串, 求出他們的LCS。現在我們把問題弄得難一點, 給三個字串, 請求出他們的LCS長度為多少?

		j	0	1	2	3	4	5	6	7
i		y	a	e	d	f	h	r	z	
	0	x	0	0	0	0	0	0	0	0
1	a	0	1	1	1	1	1	1	1	1
2	b	0	1	1	1	1	1	1	1	1
3	c	0	1	1	1	1	1	1	1	1
4	d	0	1	1	2	2	2	2	2	2
5	g	0	1	1	2	2	2	2	2	2
6	h	0	1	1	2	2	3	3	3	3
7	x	0	1	1	2	2	3	3	3	3
8	y	0	1	1	2	2	3	3	3	3

<http://lcs-demo.sourceforge.net/> (<http://lcs-demo.sourceforge.net/>)

兩個字串解法

$$C[i, j] = \begin{cases} 0, & \text{if } i = 0 \text{ or } j = 0 \\ C[i - 1, j - 1] + 1, & \text{if } i, j > 0 \text{ and } x_i = y_j \\ \text{Max}(C[i, j - 1], C[i - 1, j]), & \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases}$$

三個字串解法

$$C[i, j, k] = \begin{cases} 0, & \text{if } i = 0 \text{ or } j = 0 \text{ or } k = 0 \\ C[i - 1, j - 1, k - 1] + 1, & \text{if } i, j, k > 0 \text{ and } x_i = y_j = z_k \\ \text{Max}(C[i - 1, j, k], C[i, j - 1, k], C[i, j, k - 1]), & \text{if } i, j, k > 0 \text{ and } x_i \neq y_j \neq z_k \end{cases}$$

## 輸入

每個測資檔僅包含一筆測資，每筆測資為三列字串。字串內容為英文字母或數字，大小寫的英文字母不同，每列最多有100個字元。要注意不能先找兩個字串的LCS再跟第三個找，因為不保證最佳解出現在前兩個字串的LCS中。

## 輸出

對輸入的每組測試資料，輸出它們最長共同子序列的長度。

## 範例輸入輸出

範例輸入 I

```
1 | abcd bceea
2 | cabdefga
3 | dcea
```

範例輸出 I

```
1 | 3
```

範例輸入 II

```
1 | abe
2 | acb
3 | babcd
```

範例輸出 II

```
1 | 2
```

## P 錢幣

你有  $n$  個具有特定價值的錢幣。您的任務是找到您可以使用這些錢幣創建的所有金額。

### 輸入

第一列的數字  $n$  代表錢幣的數量， $1 \leq n \leq 100$ ，第二列起為  $n$  個整數  $x_1, x_2, \dots, x_n$  為錢幣的價值， $1 \leq x_i \leq 1000$ 。以空白隔開的正整數。

### 輸出

首先輸出一個整數  $k$ ：不同貨幣金額的數量。在此之後，按升序(遞增)，輸出所有可能的總和，以空白隔開整數。

### 範例輸入輸出

範例輸入 I

1	4
2	4 2 2 5

範例輸出 I

1	9
2	2 4 5 6 7 8 9 11 13

## Q 果子堆合併

一開始果子自成一堆一堆。每當要合併兩堆果子時，合併操作的成本為兩堆果子數量之總和。試問：將所有果子合併成一堆，其最小成本為多少？

每一次合併，你可以把兩堆果子合併到一起，消耗的體力等於兩堆果子的重量之和。可以看出，所有的果子經過 $n-1$ 次合併之後，就只剩下一堆了。多多在合併果子時總共消耗的體力等於每次合併所耗體力之和。

因為還要花大力氣把這些果子搬回家，所以你在合併果子時要盡可能地節省體力。假定每個果子重量都為1，並且已知果子的種類數和每種果子的數目，你的任務是設計出合併的次序方案，使你耗費的體力最少，並輸出這個最小的體力耗費值。

例如有3種果子，數目依次為1，2，9。可以先將1、2堆合併，新堆數目為3，耗費體力為3。接著，將新堆數目為3與原先的第三堆果子數量為9合併，又得到新的一堆，數目為12，耗費體力為12。所以你總共耗費體力 $=3+12=15$ 。可以證明15為最小的體力耗費值。

### 輸入

輸入有多筆測試資料，每筆佔兩列。測資第一列給定一正整數  $n$  ( $1 \leq n \leq 30000$ )，代表有  $n$  種果子堆。第二列給定  $n$  個正整數，用空格分隔，第  $i$  個整數  $a_i$  ( $1 \leq a_i \leq 20000$ )，分別代表各果子堆的數量。

### 輸出

每組輸出包括一列，這一系列只包含一個整數，也就是最小的體力耗費值。輸入正整數保證這個值小於 $2^{31}$ 。

### 範例輸入輸出

範例輸入 I

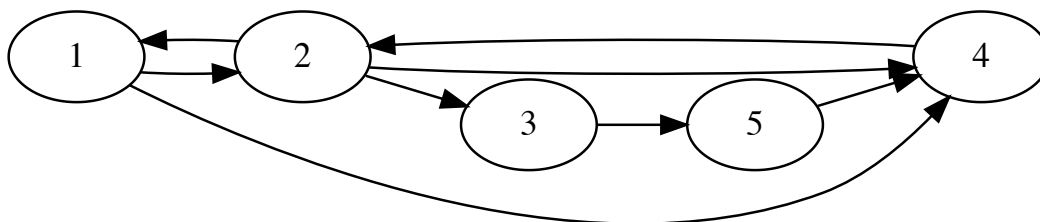
```
1 | 3
2 | 1 2 9
```

範例輸出 I

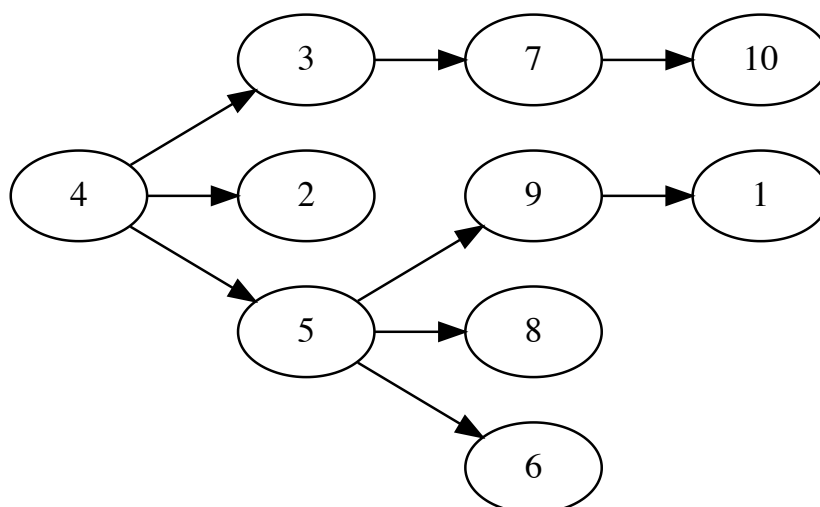
```
1 | 15
```

## R 條條大路通羅馬

條條大路通羅馬，有 $N$ 個城市 $M$ 條道路， $M$ 條道路是有方向性的。



- 問上圖：3城市是否可到達1城市，是否存在路徑可由3到1？Yes  
3 -> 5 -> 4 -> 2 -> 1



- 問上圖：3城市是否可到達1城市，是否存在路徑可由3到1？No

## 輸入

一筆測資輸入，一筆輸入有兩個正整數  $N, M$  ( $N \leq 800, M \leq 10000$ )，代表有 $N$ 個城市 $M$ 條道路， $M$ 條道路是有方向性的。接下來有 $M$ 列，每列有2個正整數  $a, b$  ( $1 \leq a, b \leq N$ )，代表 $a$ 城市可以到 $b$ 城市。最後一列有兩個正整數  $A, B$  ( $1 \leq A, B \leq N$ )，代表 $A$ 城市和 $B$ 城市。以空白隔開的正整數。

## 輸出

如果 $A$ 城市在有方向性的道路，可否到達 $B$ 城市，可以到達則輸出Yes，不行到達則輸出 No。

## 範例輸入輸出

### 範例輸入 I

1	5 8
2	1 2
3	1 4
4	2 1
5	2 3
6	2 4
7	3 5
8	4 2
9	5 4
10	1 3

### 範例輸出 I

1	Yes
---	-----

### 範例輸入 II

1	10 9
2	4 3
3	4 2
4	3 7
5	4 5
6	7 10
7	5 9
8	5 8
9	5 6
10	9 1
11	1 3

### 範例輸出 II

1	No
---	----