

**弘光科技大學 智慧科技應用系**  
**程式設計能力檢定題庫**

# 目錄

解答須知.....	1
基礎題.....	2
Q_1_1 薪資計算.....	3
Q_1_2 百貨公司購物折扣計算.....	4
Q_1_3 計算複利.....	5
Q_1_4 辨識阿姆斯壯數.....	6
Q_1_5 十進位鏡反數.....	7
Q_1_5-1 二進位鏡反數.....	8
Q_1_6 二進位數位元為 1 的個數與位置.....	9
Q_1_7 分針與時針之間的銳角夾角的度數.....	10
Q_1_8 分數級數的和.....	11
Q_1_9 級數的和.....	12
Q_2_1 所有因數.....	13
Q_2_2 辨識完美數.....	14
Q_2_3 辨識質數.....	15
Q_2_4 辨識醜數.....	16
Q_2_5 最大公因數.....	17
Q_2_6 互質.....	18
Q_3_1 三角形辨識.....	19
Q_3_2 轉換百分制成績至五等第制.....	20
Q_3_3 判斷閏年.....	21
Q_4_1 最大值.....	22
Q_4_2 不重複輸出經排序過的陣列之元素.....	23
Q_4_3 反向輸出.....	24
Q_4_5 交換陣列之前半與後半.....	25
Q_5_1 一週七天的英文單字查詢.....	26
Q_6_1 英文大小寫字母互換.....	27
Q_6_2 反轉輸出.....	28
進階題.....	29
QA_1_1 級數的和.....	30
QA_1_2 檢查身分證字號之有效性.....	31
QA_1_3 迴文數.....	33
QA_4_1 不重複輸出陣列的元素.....	34
QA_4_2 陣列中不重複的元素之個數.....	35
QA_5_1 查表 (table lookup).....	36
QA_5_2 英漢字典查詢.....	37

QA_5_3 統一發票對獎 .....	38
QA_5_4 利用人名查詢電話 .....	40
QA_5_5 利用電話查詢人名 .....	41
QA_6_1 英文日期格式轉換數字格式 .....	42
QA_7_1 平面上點集之最近對(closest pair) .....	43
QA_7_2 平面點集合的極點集 .....	44
QA_7_3 刪除平面上被支配的點座標 .....	45
QA_7_4 不被包含的區間集 .....	46
QA_8_1 合併遞增陣列 .....	47
QA_8_2 字典順序(lexicographic order)比較 .....	48
QA_8_3 排序函式測試 .....	49
QA_8_4 第 $k$ 大的數 .....	50
QA_8_5 等價陣列之判斷 .....	51
QA_9_1 陣列中和為零的段落 .....	53
QA_9_2 陣列等值首尾和位置 .....	54
QA_9_3 整數陣列中差異最小的兩數 .....	55
QA_9_4 最長等值子陣列之長度與值 .....	56
QA_9_5 最長等值子陣列之位置 .....	57
QA_10_1 分屬不同的陣列的兩元素之和 .....	58
QA_10_2 分屬不同的遞增陣列的兩元素之和 .....	59
QA_10_3 兩陣列的等值對 .....	60
QA_10_4 兩陣列的距離 .....	61
QA_10_5 求兩遞增陣列的等值對 .....	62
QA_10_6 求兩遞增陣列的距離 .....	63
QA_10_7 集合的交集 .....	64
QA_10_8 尋找三陣列的共同元素 .....	65

## 解答須知

1. 所有輸出入皆為標準輸出入 (standard input/output)，故皆為文字格式之輸出入。請勿讀取檔案。
2. 請直接讀取輸入，切勿輸出輸入提示語 (prompt)。
3. 題目如果明示輸入的格式或數值大小，輸入一定合乎題意說明，不必設計防呆程式碼檢查輸入是否合乎題目說明。
4. 題目若未說明輸入的長度，請利用 EOF 符號作為輸入結束的訊號。
5. 輸出必須確實依照題目要求，多一個字母或少一個字母都不行。
6. 不加入中文註解，避免編譯程式與中文字碼不相容導致錯誤。

# 基礎題

## Q\_1\_1 薪資計算

假設某便利商店的工讀生的月薪資，可以依照下列方式計算：

60 個小時之內，每小時 75 元

61 ~ 75 個小時，以 1.25 倍計算

76 個小時以後以 1.75 倍計算

例如，如果工作時數為 80 小時，則薪資

$60 * 75 + 15 * 75 * 1.25 + 5 * 75 * 1.75 = 6562.5$  元。

請撰寫一個程式，輸入工作時數(為一整數)，輸出該工作時數可領之薪資(捨去小數部分)。

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 1000000$ 。

第二列為  $n$  個正整數，數與數之間以空白字元隔開。一個正整數為某一個員工的工作時數。

### 輸出：

共  $n$  列。依序針對輸入的第二列的每個工作時數，輸出一列。內容為該工作時數可以領取的薪資 (捨去小數部分)。

### 輸入範例：

```
3
61 15 77
```

### 輸出範例：

```
4593
1125
6168
```

## Q\_1\_2 百貨公司購物折扣計算

百貨公司正在周年慶，消費者消費滿千元(含)以上，打八折優待，未滿千元沒有折扣。寫個程式協助消費者計算折扣後購物金額。

### 輸入：

兩列：

第一列為購買筆數  $n$ ， $0 < n \leq 1000000$ 。

第二列為  $n$  筆購買金額，單位為元。數與數之間以空白字元隔開。

### 輸出：

$n$  列：

依序針對輸入的第二列個的每一筆購物金額  $M$  輸出其折扣後的金額 (捨去小數部份) 於一列。

### 輸入範例：

```
4
999 1500 3000 1000
```

### 輸出範例：

```
999
1200
2400
800
```

## Q\_1\_3 計算複利

撰寫一個程式，可計算  $n$  組定期定額投資收益結果，利率採複利計算。假設每年固定存入金額為  $X$  元，年複利率為  $R\%$ ，第  $Y$  年後領回。

例如：每年固定存入金額為 100000 ( $X$ ) 元，存 3 ( $Y$ ) 年，若是複利率為 2.25 ( $R$ )%，3 ( $Y$ ) 年後的本利和計算如下：

第一年初存入 1000，本金=1000

第二年初，前一年本利 =  $1000 * (1 + 2.25\%) = 1022.5$ ，捨去小數部分得 1022 之後再加上存入的 1000，合計 2022

第三年初，前一年本利 =  $2022 * (1 + 2.25\%) = 2067.495$ ，捨去小數部份加上存入的 1000，合計 3067

第三年尾期滿後合計領回的本利和 =  $3067 * (1 + 2.25\%) = 3136$

### 輸入：

$n + 1$  列：

第一列為正整數  $n$ ，表示要計算  $n$  組定期定額投資收益結果。

第二列起每列有 3 個數字，數與數之間請以空白符號隔開，第一個數字為正整數  $X$  (每年存入的金額)，第二個數字為正整數  $Y$  (存入  $Y$  年後領回)，第三個數字為年複利率  $R$  (非整數，亦即有小數部份)。

### 輸出：

輸出為  $n$  列，針對輸入第二列起每列的  $X$ 、 $Y$ 、 $R$  輸出以複利計算期滿後領回的金額 (捨去小數部份)，針對輸入每列的輸出之間以一個空白字元隔開，最後一個輸出之後輸出一個跳行字元 ('\n')。

### 輸入範例：

```
3
10000 1 2.25
10000 2 2.25
10000 15 1.67
```

### 輸出範例：

```
10225 20680 171681
```



## Q\_1\_4 辨識阿姆斯壯數

**阿姆斯壯數**是指一個十進位的三位數的自然數，其百位數、十位數、以及個位的立方和恰等於該數的本身。

例如：153 是一個**阿姆斯壯數**，因為  $153 = 1^3 + 5^3 + 3^3$ 。

### 輸入：

不定個數的自然數，數與數之間以空白字元 (white space) 隔開。最後以 0 代表輸入結束。

### 輸出：

依序針對輸入的每一個自然數  $k$ ，輸出一列：假如是阿姆斯壯數，輸出 “Yes, it’s an Armstrong number!”，否則輸出 “No, it isn’t an Armstrong number!”。

### 輸入範例：

109 407 371 871 153 370 0

### 輸出範例：

No, it isn’t an Armstrong number!

Yes, it’s an Armstrong number!

Yes, it’s an Armstrong number!

No, it isn’t an Armstrong number!

Yes, it’s an Armstrong number!

Yes, it’s an Armstrong number!

## Q\_1\_5 十進位鏡反數

一個正整數的**十進位鏡反數**(decimal mirror-image number)為原來數字的十進位表示法由右至左反轉後的十進位數，例如 19832 的鏡反數為 23891。

### 輸入：

不定個數的正整數，數與數之間以空白字元 (white space) 隔開。最後以 0 代表輸入結束。

### 輸出：

依序輸出輸入的每一個正整數之十進位鏡反數，數與數之間以兩個空白字元隔開。注意若為可被十整除的數，輸出其十進位鏡反數只要輸出至最高不為 0 的位數。例如輸出 900 的十進位鏡反數為 9。

### 輸入範例：

100 23 69 900 78 89 90 55000 98 96728 0

### 輸出範例：

1 32 96 9 87 98 9 55 89 82769

## Q\_1\_5-1 二進位鏡反數

一個數的二進位表示法的位元位置編號由右(LSB)至左(MSB)由 0 開始由小到大依序編號。例如整數 61 (十進位表示法) 其八位元的二進位表示法 00100101。由於位元為 1 的最大位置編號為 5，稱 61 為二進位的 6 位數。同理稱 61 為十進位的 2 位數。一個正整數的**二進位鏡反數**(binary mirror-image number)為原來數字的二進位表示將位置編號由 0 開始至位元為 1 且位置編號最大的位元反轉之後的二進位數，例如 00101001 的二進位鏡反數為 00100101。

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 1000$ ，表示共有  $n$  個十進位正整數需要計算其二進位鏡反數。

第二列為  $n$  個十進位正整數，數與數之間以空白字元隔開。

### 輸出：

依序輸出輸入的第二列的  $n$  個**正整數之二進位鏡反數之十進位表示法**，數與數之間以兩個空白字元隔開。輸出最後一個鏡反數之後輸出一個換行字元 ('\n')。例如十進位數 126 之二進位表示為 1111101，其二進位鏡反數為 1011111，為十進位數 95。又例如 256 之二進位表示為 10000000，其二進位鏡反數為 00000001，為十進位數 1。

### 輸入範例：

```
3
95 256 125
```

### 輸出範例：

```
125 1 95
```

## Q\_1\_6 二進位數位元為 1 的個數與位置

寫一個程式輸入十進位數，輸出該數的二進位表示法共有幾個位元為 1 以及哪些位元為 1。一數的二進位表示法之位元位置編號由右至左由 0 開始由小到大。例如十進位數 61，其二進位表示法為 111101，共有 5 個位元為 1，為 1 的位元位置有 0、2、3、4 與 5。

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 1000$ 。

第二列為  $n$  個十進位正整數，數與數之間以空白字元隔開。

### 輸出：

$n$  列。針對輸入第二列的每一個數字  $I$  依序輸出一列，格式如下：

假設  $I$  的二進位表示有  $k$  個位元為 1，輸出

$k \{h_1 h_2 \dots h_k\}$

其中  $h_1, h_2, \dots, h_k$  依序由小至大為  $I$  的二進位表示法的所有位元為 1 的位置。位元位置與位元位置之間以空白字元隔開。

### 輸入範例：

2

61 15

### 輸出範例：

5 {0 2 3 4 5}

4 {0 1 2 3}

## Q\_1\_7 分針與時針之間的銳角夾角的度數

撰寫一個程式輸入時間(小時：分鐘)，計算在該時間時鐘上依順時鐘方向從分針到時針之間的夾角的度數。例如時間 (小時:分鐘) 為 3:15 (3 點 15 分) 時，時針與分針之銳角夾角的度數為 7.5 度。

### 輸入：

$n + 1$  列：

第一列為整數  $n$ 。

第二列起每列兩個數字，數字與數字之間以空白字元隔開。每列的第一個數字為小時數 (hour)，第二個數為分鐘數 (minute)。例如範例輸入的第二列表示時間為 3 時 15 分。

### 輸出：

$n$  列：

依序針對輸入第二列起每列的時間輸出該時間時鐘上依順時鐘方向從分針到時針之間的夾角的度數於一列。計算至小數點後一位。**請注意，依順時鐘方向從分針到時針的夾角不一定是銳角。**

### 輸入範例：

```
2
3 15
4 30
```

### 輸出範例：

```
7.5
315.0
```

## Q\_1\_8 分數級數的和

請設計一個程式輸入正整數  $n$ ，輸出下列級數的值：

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots \pm \frac{1}{n}$$

**輸入：**

共  $t+1$  個數字，數字與數字之間以空白字元隔開。第一個數字為  $t$ 。

**輸出：**

針對輸入的第二個數字起，依序對每一個數字  $n$ ，輸出一列，內容為級數

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \dots \pm \frac{1}{n}$$

的值。輸出至小數點後六位。

**輸入範例：**

2 7 10

**輸出範例：**

0.759524

0.645635

## Q\_1\_9 級數的和

試撰寫一程式計算級數  $(1 + 2) + (2 + 4) + (3 + 6) + \dots + (n + 2 * n)$  的和。

**輸入：**

$t + 1$  個數字，數字與數字之間以空白字元隔開。第一個數字為  $t$ 。

**輸出：**

$t$  列。針對輸入的第二個數字起每一個數字  $n$ ，依序輸出級數  $(1 + 2) + (2 + 4) + (3 + 6) + \dots + (n - 1 + 2 * (n - 1)) + (n + 2 * n)$  的和於一列。

**輸入範例：**

2 7 9

**輸出範例：**

84

135

## Q\_2\_1 所有因數

撰寫一程式輸入  $t$  個正整數，針對輸入的每一個正整數  $N$ ，依由小到大的順序輸出其所有的因數。

**輸入：**

$t+1$  個正整數，數與數之間以空白字元隔開。第一個數為  $t$ 。

**輸出：**

$t$  列：

依序針對輸入的第二個數字起的每一個正整數  $N$ ，依由小到大的順序輸出其所有的因數於一行。數字與數字之間以一個空白字元隔開。

**輸入範例：**

2 20 79

**輸出範例：**

1 2 4 5 10 20

1 79



## Q\_2\_2 辨識完美數

正整數  $n$  的真因數是指比  $n$  小的因數。假設正整數  $n$  之真因數 (proper factor) 的總和等於  $n$ ，則稱之為**完美數** (perfect number)，例如以 6、28、496 等皆是完美數：

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

$$496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$$

但是 49、69 等就不是完美數。49 的真因數有 1 和 7，但是

$$49 \neq 1 + 7 = 8$$

因此 49 不是完美數。同理，69 的真因數有 1、3 和 23，但是

$$69 \neq 1 + 3 + 23 = 27$$

因此 69 不是完美數。

試撰寫一程式，辨識**完美數**。

### 輸入：

不定個數的自然數，數與數之間以空白字元 (white space) 隔開。最後以 0 代表輸入結束。

### 輸出：

針對輸入的每一個自然數  $k$  輸出一列，假如是完美數，輸出內容為 “Yes, it’s a perfect number!”，否則輸出內容為 “No, it isn’t a perfect number!”。

### 輸入範例：

496 407 371 6 28 871 0

### 輸出範例：

Yes, it’s a perfect number!

No, it isn’t a perfect number!

No, it isn’t a perfect number!

Yes, it’s a perfect number!

Yes, it’s a perfect number!

No, it isn’t a perfect number!

## Q\_2\_3 辨識質數

大於 1 的整數，除了 1 與本身以外沒有其他因數，稱之為『質數』(prime number)。換言之，質數僅被 1 及本身整除。因此，2、3、5、7、11....等均稱為質數。而 33 除了 1 與 33 之外還可被 3 與 11 整除，則非質數。試撰寫一程式，辨識質數。

### 輸入：

不定個數的自然數，數與數之間以空白字元 (white space) 隔開。最後以 0 代表輸入結束。所有數皆小於 1000000。

### 輸出：

針對輸入的每一個自然數  $N$ ，輸出一列。假如  $N$  是質數，輸出內容為 “Yes, it’s a prime number!”，否則輸出內容為 “No, it isn’t a prime number!”。

### 輸入範例：

71 408 371 191 131 875 0

### 輸出範例：

Yes, it’s a prime number!

No, it isn’t a prime number!

No, it isn’t a prime number!

Yes, it’s a prime number!

Yes, it’s a prime number!

No, it isn’t a prime number!

## Q\_2\_4 辨識醜數

醜數(Ugly number)的定義如下：

1. 醜數為自然數。
2. 1 為醜數。
3. 質因數只有 2、3 或 5 的的自然數是醜數。
4. 除了上述兩種之外的自然數皆不是醜數。

根據上述定義，1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15 等皆為醜數。12 是醜數是因為他的值因數只有 2 和 3。但是 14 不是醜數，因為 7 是 14 的質因數，因此 14 的質因數不是只有 2、3 或 5。

註：如果你對題意不解，請複習國中數學了解何謂自然數與質因數。

### 輸入：

不定個數的自然數，數與數之間以白色空白字元 (white space) 隔開。最後以 0 代表輸入結束。

### 輸出：

針對輸入的每一個大於 0 的自然數  $k$ ，假如是醜數，輸出 “Yes, it’s an ugly number!\n”，否則輸出 “No, it isn’t an ugly number!\n”。

### 輸入範例：

10 99 15 1080 7560 0

### 輸出範例：

Yes, it’s an ugly number!  
No, it isn’t an ugly number!  
Yes, it’s an ugly number!  
Yes, it’s an ugly number!  
No, it isn’t an ugly number!

## Q\_2\_5 最大公因數

求兩個自然數的最大公因數是國中學生的數學題。戴寶在國中時期對數學不感興趣，錯過了學習輾轉相除法的時機。如今他發憤圖強希望可以練習計算兩數的最大公因數。老師給了他  $n$  對的自然數，要他逐一練習計算它們的最大公因數。戴寶希望你幫忙他寫個程式算出每對自然數的最大公因數，協助檢驗他的計算結果是否正確。

### 輸入：

$n + 1$  列。

第一列為整數  $n$ ， $0 < n \leq 100000$ 。

第二列起每列各有兩個自然數，數與數之間以空白字元隔開，共有  $n$  列。

### 輸出：

針對輸入的第二列起每列輸出一列，內容為該列的兩自然數之最大公因數。

### 輸入範例：

```
5
11 22
6 8
125 45
81 36
83 249
```

### 輸出範例：

```
11
2
5
9
83
```

## Q\_2\_6 互質

兩個正整數除了 1 之外若沒有其它公因數，稱他們為**互質**。例如 6 與 29 兩數除了 1 之外若沒有其它公因數，因此互質；反之 81 與 36 除了 1 之外，3 與 9 皆為它們的公因數，因此不為互質。

### 輸入：

$n + 1$  列。

第一列為整數  $n$ ， $0 < n \leq 100000$ 。

第二列起每列各有兩個正整數，數與數之間以空白字元隔開。

### 輸出：

依序針對輸入的第二列起，每列的一對數字，輸出一列結果。如果兩數互質，輸出內容為“**Yes**”；否則輸出內容為兩數的最大公因數。

### 輸入範例：

5

11 22

6 29

125 43

81 36

83 249

### 輸出範例：

11

Yes

Yes

9

83

## Q\_3\_1 三角形辨識

三角形的任兩邊長之和大於第三邊長。三個數值  $a$ 、 $b$  和  $c$  若任兩數和大於第三數，則可構成一個三角形的三個邊長。一個三角形若三邊皆等長，稱為**正三角形**；若有兩邊等長，稱為**等腰三角形**；若任兩邊皆不等長，稱為**一般三角形**。顯然正三角形也屬於等腰三角形。寫一個程式辨識三個整數  $a$ 、 $b$  和  $c$  是否構成三角形；如果構成三角形，是屬於正三角形、等腰三角形或是一般三角形？

### 輸入：

輸入共  $n+1$  列。第一列為正整數  $n$ ， $0 < n \leq 1000$ 。第二列起每列依序由三個整數  $a$ 、 $b$ 、 $c$  組成，數與數之間以空白字元隔開。

### 輸出：

針對輸入的第二列起每列的三個整數，輸出一列文字。每列的三個整數若不構成一個三角形的三個邊長，則輸出“非三角形”；若構成一個正三角形的三個邊長，則輸出“正三角形”；若構成一個等腰三角形的三個邊長，則輸出“等腰三角形”；若構成一個一般三角形的三個邊長，則輸出“一般三角形”。請注意若為正三角形不得輸出“等腰三角形”。

### 輸入範例：

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

### 輸出範例：

```
非三角形
等腰三角形
一般三角形
正三角形
```

## Q\_3\_2 轉換百分制成績至五等第制

美國學校的成績為五等第制。五等第制與我們的百分制之間的關係如下：

90 分以上(含) 為 A

80 分以上(含)至 90 分以下為 B

70 分以上(含)至 80 分以下為 C

60 分以上(含)至 70 分以下為 D

60 分以下為 F

請寫一個程式協助老師判斷全部學生成績的等第。

### 輸入：

兩列。

第一列為正整數  $n$ ， $0 < n \leq 10000$ ，表示共有  $n$  個成績需要判斷其等第。

第二列為  $n$  個成績，每個成績皆為 0 至 100 之間的整數，成績與成績之間以空白字元隔開。

### 輸出：

依序輸出輸入的第二列的每個成績的等第，等第之間以兩個空白字元隔開，最後輸出一個換行字元 ('\n')。

### 輸入範例：

10

100 23 69 90 0 78 89 90 55 98

### 輸出範例：

A F D A F C B A F A

## Q\_3\_3 判斷閏年

閏年的二月份為 29 天。非閏年的二月份為 28 天。西元元年至西元 3199 年之間的西元年，可為四整除者當中除了可為一百整除但不能為四百整除者之外均為閏年，其他年份皆不是閏年(每四年一閏,每百年不閏,每四百年一閏)。請寫一個程式協助戴哥判斷一個西元年份是否為閏年。

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 1000$ ，表示共有  $n$  個西元年需要判斷其是否為閏年。第二列為  $n$  個大於 0 但是小於 3200 的西元年份，年份之間以空白字元隔開。

### 輸出：

針對輸入第二列的每一個年份，依序判定其是否為閏年，若為閏年輸出 Y，否則輸出 N。針對不同份的輸出之間以一個空白字元隔開。針對最後一個年份的輸出之後輸出一個換行字元（'\n'）。

### 輸入範例：

```
9
1000 800 1200 1990 2012 2013 2000 3000 2400
```

### 輸出範例：

```
NYYNYY
```



## Q\_4\_1 最大值

已知  $A[]$  為 C 語言整數陣列，有  $n$  個元素。寫一個程式，計算陣列  $A[]$  中的元素索引從  $i$  到  $j$ ， $0 \leq i \leq j < n$ ，之間**最大的元素值**。

### 輸入：

3 列：

第一列依序為兩個正整數  $n$  與  $m$ 。

第二列依序為長度為  $n$  的 C 語言整數陣列的元素， $0 < n \leq 100000$ 。

第三列依序為  $m$  對數字，每對數字為兩個不為負的整數  $i$  與  $j$ ， $0 \leq i \leq j < n$ 。

每列輸入的數與數之間以空白字元隔開。

### 輸出：

針對輸入的第三列的每對不為負的整數  $i$  與  $j$ ， $0 \leq i \leq j < n$ ，輸出陣列  $A[]$  的索引從  $i$  到  $j$  之間**最大的元素值**。例如，

$A[] = \{2233, -2712, 1234, -2385, 1561, 158\}$ ，索引從 2 到 4 之間的數有 1234, -2385 與 1561，因此其間的**最大的元素值**為 1561。

輸出的數字與數字之間以兩個空白字元隔開。最後輸出一個換行字元('\n')。

### 輸入範例：

```
6 2
2233 -2712 1234 -2385 1561 158
0 5 2 4
```

### 輸出範例：

```
2233 1561
```

## Q\_4\_2 不重複輸出經排序過的陣列之元素

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。`intAry[]` 中元素已經依由小到大的順序排列。`intAry[]` 中的元素值不一定唯一。例如

```
intAry[] = {-8, -7, -7, -6, -2, -2, 1, 1, 1}
```

中有兩個元素的值等於-7，有三個元素的值等於1，但是只有一個元素的值等於-6或-8。稱陣列中元素值去掉重複的數值只保留一個後的數值所形成的集合為陣列的**代表數集**。例如，上列 `intAry[]` 陣列的代表數如下：

```
{-8, -7, -6, -2, 1}。
```

寫一程式讀入  $n$  個元素的 C 語言陣列，輸出它的代表數集的元素個數  $k$ ，再依由小到大的順序輸出陣列的代表數集的元素，數與數之間以一個空白字元隔開。輸出最後一個數之後輸出一個換行字元('\n')。

### 輸入：

兩列：

第一列為整數陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。

第二列為整數陣列 `intAry[]` 的  $n$  個元素；元素與元素之間以空白字元隔開。

### 輸出：

兩列：

第一列為陣列 `intAry[]` 的代表數集的元素個數。

第二列為陣列 `intAry[]` 的代表數集的元素，依由小到大的順序輸出，數與數之間以一個空白字元隔開。輸出最後一個數之後輸出一個換行字元('\n')。

### 輸入範例：

```
9
-8 -7 -7 -6 -2 -2 1 1 1
```

### 輸出範例：

```
5
-8 -7 -6 -2 1
```

## Q\_4\_3 反向輸出

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 10000$ 。

第二列為  $n$  個整數，數與數之間以空白字元隔開。

### 輸出：

反向依序輸出輸入的第二列之  $n$  個整數，數字之間以兩個空白字元格開。最後一個數字之後緊接著為一個換行字元（'\n'）。

### 輸入範例：

9

1000 800 1200 1990 2012 2013 2000 3000 1000

### 輸出範例：

1000 3000 2000 2013 2012 1990 1200 800 1000

## Q\_4\_5 交換陣列之前半與後半

### 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 100000$ 。

第二列為  $n$  個整數，數與數之間以一個空白字元隔開。

### 輸出：

先依序逐一輸出輸入的第二列的後半部分（後  $\lfloor n/2 \rfloor$  個）數字後再依序逐一輸出輸入的前半部分（ $\lceil n/2 \rceil$  個）數字。數與數之間以一個空白字元格開。最後一個數字之後以一個換行字元（'\n'）。

### 輸入範例：

9

1000 800 1200 1990 2012 2013 2000 3000 1000

### 輸出範例：

2013 2000 3000 1000 1000 800 1200 1990 2012

## Q\_5\_1 一週七天的英文單字查詢

寫個程式輸入“星期一”、“星期二”、“星期三”、“星期四”、“星期五”、“星期六”、“星期日”等其中之一，輸出其英文單字。例如輸入“星期一”，輸出“Monday”；輸入“星期二”，輸出“Tuesday”；輸入“星期三”，輸出“Wednesday”；輸入“星期四”，輸出“Thursday”；輸入“星期五”，輸出“Friday”；輸入“星期六”，輸出“Saturday”；輸入“星期日”，輸出“Sunday”。

### 輸入：

$n + 1$  列：

第一列為整數  $n$ ， $0 < n \leq 100000$ 。

第二列起每列內容為“星期一”、“星期二”、“星期三”、“星期四”、“星期五”、“星期六”、“星期日”等七者之一。

### 輸出：

依序針對輸入的第二列起的每列輸入，輸出其英文單字於一列，若輸入非星期一至星期日其中之一，輸出“Out of range!”。

### 輸入範例：

```
5
星期一
星期日
星期六
星期日
星期八
```

### 輸出範例：

```
Monday
Sunday
Saturday
Sunday
Out of range!
```

## Q\_6\_1 英文大小寫字母互換

寫一個程式，輸入一系列的英文字母，將其中的小寫英文字母轉成對應的大寫英文字母，大寫英文字母轉成對應的小寫英文字母之後輸出。請注意非 26 個英文字母的其他字符不予以變更。

### 輸入：

一列由英文字元組成的字串。

### 輸出：

將其中的小寫英文字母轉成對應的大寫英文字母，大寫英文字母轉成對應的小寫英文字母之後輸出。非 26 個英文字母的其他字元不予以變更。

### 輸入範例：

Good morning

### 輸出範例：

gOOD MORNING

## Q\_6\_2 反轉輸出

寫一個程式，輸入一系列長度不超過 255 字元的英文文字將其反轉輸出。

### 輸入：

一系列長度不超過 255 字元的英文文字。

### 輸出：

將輸入反轉後輸出，亦即先輸入的字母後輸出。

### 輸入範例：

I love you!

### 輸出範例：

!uoy evol I

## 進階題



## QA\_1\_1 級數的和

請撰寫一個程式來計算下面的數學式的值：

$$\sum_{k=1}^n \frac{x^k}{k!} = \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

**輸入：**

兩列：

第一列為正整數  $m$ ， $0 < m \leq 100000$ 。

第二列為  $2m$  個整數，數與數之間以一個空白字元隔開。每兩個數一組，第一與第二個數為第一組；第三與第四個數為第二組；以此類推共得  $m$  組。每組數字的第一個數為  $x$ ，第二個數為  $n$ 。

**輸出：**

依讀入的順序逐一輸出每組數字計算結果於一列。請用 `double` 型態的變數來計算。輸出小數點後六位數。每組數字輸出一個值，**兩組的輸出之間以兩個空白字元隔開**。最後一組的輸出之後輸出一個換行字元（`'\n'`）。

**輸入範例：**

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

**輸出範例：**

```
1.000000 137.307168 6.000000 18.412500 18.846429
```

## QA\_1\_2 檢查身分證字號之有效性

身分證字號之格式為一個大寫英文字母後接 9 位阿拉伯數字。每個英文字母各代表一個縣市(五都尚未成立之前)；另每個大寫英文字母各有一個兩位數的代碼如下表：

A	台北市	10	J	新竹縣	18	S	高雄縣	26
B	台中市	11	K	苗栗縣	19	T	屏東縣	27
C	基隆市	12	L	台中縣	20	U	花蓮縣	28
D	台南市	13	M	南投縣	21	V	台東縣	29
E	高雄市	14	N	彰化縣	22	W	金門縣	32
F	台北縣	15	O	新竹市	35	X	澎湖縣	30
G	宜蘭縣	16	P	雲林縣	23	Y	陽明山	31
H	桃園縣	17	Q	嘉義縣	24	Z	馬祖	33
I	嘉義市	34	R	台南縣	25			

請撰寫一程式可驗證使用者所輸入的身分證字號是否為有效證號。驗證規則公式如下：

首先依據上表將身分證字號的大寫英文字母轉換成十進位的兩位數，之後接上原來的 9 位數形成 11 位數的代碼。再將 11 位數的代碼的每位數由左至右依序分別乘上 1、9、8、7、6、5、4、3、2、1、1 後相加所得的數字若為 10 的倍數，則為有效的身分證號碼；否則為無效的身分證號碼。例如: B120863158，B 轉換成十進位的兩位數 11 之後接上原來的 9 位數形成 11 位數的代碼：11120863158；再將 11120863158 的每位數由左至右依序分別乘上 1、9、8、7、6、5、4、3、2、1、1 後相加：

$$1 \times 1 + 1 \times 9 + 1 \times 8 + 2 \times 7 + 0 \times 6 + 8 \times 5 + 6 \times 4 + 3 \times 3 + 1 \times 2 + 5 \times 1 + 8 \times 1 \\ = 1 + 9 + 8 + 14 + 40 + 24 + 9 + 2 + 5 + 8 = 120$$

所得的數字為 10 的倍數，故為有效的身分證號碼。

## 輸入：

$n + 1$  列：

第一列為正整數  $n$ ， $0 < n \leq 1000$ 。

第二列起共  $n$  列，每列為一個身分證字號。

## 輸出：

針對輸入的每列身分證字號輸出一列。身分證字號若為有效則輸出“有效之身分證字號”；否則輸出“無效之身分證字號”。

## 輸入範例：

```
5
G108276816
K249483438
P1000098765
L141103806
L36874321
```

## 輸出範例：

```
有效身份證字號
有效身份證字號
無效身份證字號
有效身份證字號
無效身份證字號
```

## QA\_1\_3 迴文數

**迴文數** (palindrome)是指一個從左到右與從右到左的讀法相同的十進位自然數。例如:12321 就是迴文數，1234 則不是。

### 輸入：

不定個數的自然數，數與數之間以空白字元 (white space) 隔開。最後以 0 代表輸入結束。

### 輸出：

$n$  列：

依序針對輸入的每一個自然數  $N$  輸出 “Yes” 或者 “No” 於一列。假如  $N$  為迴文數則輸出 “Yes” 否則輸出 “No”。

### 輸入範例：

151 231 1221 0

### 輸出範例：

Yes  
No  
Yes

## QA\_4\_1 不重複輸出陣列的元素

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。寫一程式依序從索引為 0 開始逐一輸出 `intAry[]` 陣列之元素，一個元素的值如果與已經輸出過的元素的值相等則不再輸出。

### 輸入：

兩列：

第一列為陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。

第二列為陣列 `intAry[]` 的  $n$  個元素；元素與元素之間以空白字元隔開。

### 輸出：

一列：

依序從索引為 0 開始逐一輸出 `intAry[]` 陣列之元素，一個元素的值如果與已經輸出過的元素的值相等則不再輸出。輸出最後一個元素之後輸出一個換行字元（'\n'）。

### 輸入範例：

9

1 -2 1 -7 -6 -7 1 -8 -2

### 輸出範例：

1 -2 -7 -6 -8

## QA\_4\_2 陣列中不重複的元素之個數

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。寫一程式計算 `intAry[]` 陣列中不重複的元素個數  $k$ 。定義 `intAry[]` 中不與任何其它元素相等的元素為 `intAry[]` 的不重複的元素。假設

`intAry[] = {1, -2, 1, -7, 6, -7, 1, 8, -2}`，因為有兩個(含)以上的元素的值為 1，故 1 不是 `intAry[]` 的不重複的元素；同理，-2 和 -7 也都不是。反之，-6 與 8 兩個元素值皆不與 `intAry[]` 中任何其它元素值相等故為 `intAry[]` 的不重複的元素。`intAry[]` 的不重複的元素共有 -6 與 8 兩個，因此  $k=2$ 。

輸出  $k$  之後再依在陣列中之索引由小到大的順序，輸出 `intAry[]` 陣列中不重複的元素，輸出最後一個元素之後輸出一個換行字元 ('\n')。

### 輸入：

兩列：

第一列為陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。

第二列為陣列 `intAry[]` 的  $n$  個元素；元素與元素之間以空白字元隔開。

### 輸出：

兩列：

第一列為不重複的元素個數  $k$ 。

第二列依在陣列中之索引由小到大的順序，輸出 `intAry[]` 陣列中不重複的元素，元素與元素之間以兩個空白字元隔開。最後一個元素輸出之後，輸出一個換行字元 ('\n')。

### 輸入範例：

9

1 -2 1 -7 -6 -7 1 8 -2

### 輸出範例：

2

-6 8

## QA\_5\_1 查表 (table lookup)

查表是程式設計的基本技巧。可以視為最簡單形式的搜尋。請寫個程式從一個線性表格中找出第  $k$  欄的資料輸出。

### 輸入：

輸入共有  $n+2$  列。第一列有兩個正整數  $n$  與  $m$ ， $0 < n \leq 10000$ ， $0 < m \leq 10000$ 。以一個空白字元隔開。第二列到第  $n+1$  列共  $n$  列，每列皆為一句長度不超過 128 字元的英文句子。第  $n+2$  列為  $m$  個整數  $k$ ， $1 \leq k \leq n$ ，數與數之間以一個空白字元隔開。

### 輸出：

依序針對第  $n+2$  列的每一個整數  $k$  輸出輸入的第  $k+1$  列。

### 輸入範例：

10 5

We are mother and daughter.

It is a dog.

Is your dog big or small?

He isn't my mom's student.

My daughter has many books.

Where do you live?

Do your students eat breakfast every day?

Does Amy like music?

Where is your teacher from?

His brother played the piano last night.

9 3 1 6 9

### 輸出範例：

Where is your teacher from?

Is your dog big or small?

We are mother and daughter.

Where do you live?

Where is your teacher from?

## QA\_5\_2 英漢字典查詢

很多初學英文的學生喜歡使用電子英文詞典。電子英文詞典的一個重要功能是輸入一個英文單字，顯示其中文字義。請寫個程式輸入英文單字，輸出其中文字義。

### 輸入：

$n + 2$ ：

第一列有兩個正整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 10000$ ，以空白字元隔開。

第二列到第  $n + 1$  列，每列包含一個英文單字及其中文字義。英文單字在前，中文字義在後，英文單字與中文字義之間以空白字元隔開。

第  $n + 2$  列為  $m$  個英文單字，單字與單字之間以空白字元隔開。

所有英文單字的字母皆為小寫。英文單字長度不超過 40 個字母。中文字義長度不超過 10 個中文字。

### 輸出：

依序針對輸入第  $n + 2$  列的每一個單字輸出輸入中該單字的中文字義於一列。如果這個單字不存在輸入中，請輸出“`It doesn't exist!`”。

### 輸入範例：

```
8 5
daughter 女兒
hungry 飢餓的
student 學生
book 書
live 居住
breakfast 早餐
teacher 老師
piano 鋼琴
teacher small breakfast piano hungry
```

### 輸出範例：

```
老師
It doesn't exist!
早餐
鋼琴
飢餓的
```



## QA\_5\_3 統一發票對獎

下表為民國 96 年 9 月、10 月的統一發票中獎號碼單。

96 年 9 月、10 月統一發票中獎號碼單	
月份	9~10 月
特獎	41292387
	同期統一發票收執聯 8 位數號碼與上列號碼相同者獎金 200 萬元
頭獎	32971009
	39376966
	50336841
	同期統一發票收執聯 8 位數號碼與上列號碼相同者獎金 20 萬元
二獎	同期統一發票收執聯末 7 位數號碼與頭獎中獎號碼末 7 位相同者各得獎金 4 萬元
三獎	同期統一發票收執聯末 6 位數號碼與頭獎中獎號碼末 6 位相同者各得獎金 1 萬元
四獎	同期統一發票收執聯末 5 位數號碼與頭獎中獎號碼末 5 位相同者各得獎金 4 千元
五獎	同期統一發票收執聯末 4 位數號碼與頭獎中獎號碼末 4 位相同者各得獎金 1 千元
六獎	同期統一發票收執聯末 3 位數號碼與頭獎中獎號碼末 3 位相同者各得獎金 2 百元

對獎規則如上列中獎號碼單所述。請撰寫程式協助使用者對獎。

## 輸入：

兩列：

第一列為正整數  $n$ ， $0 < n \leq 100000$ ，為需要對獎的統一發票張數。

第二列為  $n$  張的統一發票收執聯之號碼。每個號碼為十進位的八位數。號碼與號碼之間以一個空白字元隔開。

## 輸出：

第二列的每一個號碼，輸出其中獎金額。兩個中獎金額之間以一個換行字元（'\n'）字元隔開，最後一個號碼的中獎金額之後輸出一個換行字元（'\n'）做為結束。中特獎、頭獎、二獎、三獎、四獎、五獎、六獎以及沒中獎分別輸出 “You win 2 millions!”， “You win 200 thousands!”， “You win 40 thousands!”， “You win ten thousands!”， “You win 4 thousands!”， “You win one thousands!”， “You win two hundreds!”， “You don't win any money!”。假如一張發票同時中了兩種獎以上，輸出中獎金額最高者。

## 輸入範例：

6

50336841 56336966 41293009 65376966 82971009 41292666

## 輸出範例：

You won 200 thousands!

You won one thousands!

You won two hundreds!”

You won ten thousands!

You won 40 thousands!

You don't win any money!

## QA\_5\_4 利用人名查詢電話

一個簡單的通訊錄記錄人名與電話。這一題練習利用人名查詢電話。人名為一個最多 32 個英文字母 (A ~ Z, a ~ z) 的字串。電話由 8 個阿拉伯數字 (0 ~ 9) 組成。

### 輸入：

輸入共有  $n + 2$  列。第一列有兩個正整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 10000$ 。第二列起直至第  $n + 1$  列止，每列為通訊錄的一筆記錄。換言之這個通訊錄共有  $n$  筆記錄 (record)。每筆紀錄有兩個欄位 (field)，第一欄為人名，第二欄為第一欄人名之電話號碼，人名與電話號碼之格式如題意說明。人名與電話號碼之間以空白字元隔開。第  $n + 2$  列為  $m$  個人名，人名與人名之間以空白字元隔開。以輸入範例為例，第一列為兩個數  $n = 10$ ， $m = 5$ 。因此通訊錄有 10 筆記錄。

### 輸出：

依序針對第  $n + 2$  列的每一個人名輸出一列。如果人名在通訊錄中，輸出其電話號碼。如果這個人名不在通訊錄中，請輸出 “It doesn’t exist!\n”。請注意，一個字母的大、小寫視為同一個字母。例如 Peter 與 peter 視為相同的人名。例如範例輸入的第十二列有五個人名，故輸出五列。第十二列的第三個人名 Henry 不在通訊錄中，故輸出 “It doesn’t exist!”

### 輸入範例：

```
10 5
Peter 21888234
Mary 32153674
John 65398123
Tom 56325789
Ted 38972557
Gary 87886231
Amy 98342557
Alice 98213338
Michel 33884563
Daniel 44512398
alice Peter Henry daniel William
```

### 輸出範例：

```
98213338
21888234
It doesn't exist!
44512398
It doesn't exist!
```

## QA\_5\_5 利用電話查詢人名

實作一個簡單的通訊錄記錄人名與電話，練習利用人名查詢電話。其中人名為一個最多 32 個英文字母 (A~Z, a~z) 的字串。電話由 8 個阿拉伯數字 (0~9) 組成。

### 輸入：

輸入共有  $n+2$  列。第一列有兩個正整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 10000$ ， $n$  為通訊錄的紀錄筆數， $m$  為待查詢人名的電話號碼數。第二列起直至第  $n+1$  列止，每列為通訊錄的一筆記錄。換言之這個通訊錄共有  $n$  筆記錄 (record)。每筆錄有兩個欄位 (field)，第一欄為人名，第二欄為第一欄人名之電話號碼，人名與電話號碼之格式如題意說明。人名與電話號碼之間以空白字元隔開。第  $n+2$  列為  $m$  個電話號碼，電話號碼與電話號碼之間以空白字元隔開。以輸入範例為例，第一列為兩個數  $n=10$ ， $m=5$ 。因此通訊錄有 10 筆記錄。第  $n+2$  列有 5 個電話號碼，請利用這五個電話號碼查詢其對應的名字。

### 輸出：

依序針對第  $n+2$  列中待查詢的每一個電話號碼輸出一列。如果電話號碼在通訊錄中，輸出其對應的名字。否則請輸出 “It doesn't exist!”。例如範例輸入的第十二列有五個電話號碼，故輸出五列。第十二列的第三個電話號碼 36932678 以及第五個電話號碼 45512398 不在通訊錄中，故分別輸出 “It doesn't exist!”

### 輸入範例：

```
10 5
Peter 21888234
Mary 32153674
John 65398123
Tom 56325789
Ted 38972557
Gary 87886231
Amy 98342557
Alice 98213338
Michel 33884563
Daniel 44512398
98213338 21888234 36932678 44512398 45512398
```

### 輸出範例：

```
Alice
Peter
It doesn't exist!
daniel
It doesn't exist!
```

## QA\_6\_1 英文日期格式轉換數字格式

英文日期的文字格式為：

**日 月 年**

其中日與年皆為阿拉伯數字，月則為英文的單字；日與月之間以及月與年之間以一個空白字元隔開。例如 15 January 1993

數字日期格式則如下：

**月/日/年**

其中月、日、年皆為阿拉伯數字，日與月之間以及月與年之間以一個‘/’字元隔開。例如 1/15/1993

請撰寫程式將上述英文日期的文字格式轉換為數字日期格式。例如輸入英文日期：

15 January 1993

將其轉換為數字日期格式：

1/15/1993

**輸入：**

$n + 1$  列：

第一列為正整數  $n$ ， $0 < n \leq 1000$ ，表示共有  $n$  個英文日期要轉換為數字格式。

第二列起每列皆為一個合法的英文日期。

**輸出：**

依序輸出輸入的第二列起的  $n$  英文日期之數字格式，每個日期一列。

**輸入範例：**

2

15 January 1993

17 August 2013

**輸出範例：**

1/15/1993

8/17/2013

## QA\_7\_1 平面上點集之最近對(closest pair)

$(x, y)$  與  $(a, b)$  為平面上的點座標，其距離的計算公式如下：

$$\sqrt{(x-a)^2 + (y-b)^2}$$

寫一個程式，輸入平面上的一個點座標集合  $P$  輸出其中距離最近的兩點之點座標以及其距離。

### 輸入：

$n+1$  列：

第一列為平面上的點座標數目  $n$ ， $0 < n \leq 100000$ 。

第二列起每列為一個點座標。每列的第一個數字為其  $x$  座標，第二個數字為  $y$  座標。 $x$  座標與  $y$  座標之間以一個空白字元隔開。所有點座標之  $x$  座標與  $y$  座標皆為整數。任兩點座標之距離均大於 0。

### 輸出：

假設距離最近的兩點座標為  $(x_1, y_1)$  與  $(x_2, y_2)$  且滿足  $x_1 < x_2$  或者  $x_1 = x_2, y_1 < y_2$ 。輸出  $x_1, y_1, x_2, y_2$  以及兩點之間之距離，小數點之後最多 6 位。數與數之間以兩個空白字元隔開，最後一個數之後輸出一個換行字元 ('\n')。

### 輸入範例：

```
5
1 1
8 5
1 -2
3 6
2 4
```

### 輸出範例：

```
2 4 3 6 2.236067
```

## QA\_7\_2 平面點集合的極點集

$(x, y)$  與  $(a, b)$  為平面上的點座標。如果  $x \leq a$  且  $y \leq b$ ，則稱  $(a, b)$  支配  $(x, y)$ 。例如  $(1, -2)$  支配  $(1, -2)$ ； $(8, 5)$  支配  $(2, 4)$ ，但是  $(8, 5)$  不支配  $(3, 6)$ ； $(3, 6)$  也不支配  $(8, 5)$ 。一個平面上的點座標集合  $P$  中不被任何其它點支配的點稱為  $P$  的極點 (maximal pint)。  $P$  的所有極點形成的集合  $L$  稱為點座標集合  $P$  的極點集。寫一個程式，輸入一個平面上的點座標集合  $P$  輸出其極點集。

### 輸入：

輸入共  $n+1$  列。第一列為一正整數  $n$ ， $0 < n \leq 100000$ 。第二列起到第  $n+1$  列止共  $n$  列，每列均為點座標集合  $P$  的一個點座標。每列的第一個數為點座標的  $x$  座標，第二個數為點座標的  $y$  座標。 $x$  座標與  $y$  座標之間以一個空白字元隔開。所有點座標之  $x$  座標與  $y$  座標皆為整數。任兩點座標皆不相等。

### 輸出：

依讀入的順序逐一輸出點座標集合  $P$  的所有極點(可能不只一點)。輸出一個極點時，先輸出其  $x$  座標，再輸出其  $y$  座標； $x$  座標與  $y$  座標之間以兩個空白字元隔開。點座標與點座標之間以一個換行字元 ('\n') 隔開。輸出最後一個點座標之後以一個換行字元 ('\n') 結束。

### 輸入範例：

```
5
1 1
8 5
1 -2
2 4
3 6
```

### 輸出範例：

```
8 5
3 6
```

## QA\_7\_3 刪除平面上被支配的點座標

$(x, y)$  與  $(a, b)$  為平面上的點座標。如果  $x \leq a$  且  $y \leq b$ ，則稱點座標  $(a, b)$  支配點座標  $(x, y)$ 。例如  $(1, -2)$  支配  $(1, -2)$ ； $(8, 5)$  支配  $(2, 4)$ ，但是  $(8, 5)$  不支配  $(3, 6)$ ； $(3, 6)$  也不支配  $(8, 5)$ 。

。

### 輸入：

輸入的第一列為平面上的點座標數目  $n$ ， $0 < n \leq 100000$ 。第一列之後接著  $n$  列輸入，每列為一個點座標。每列的第一個數字為其  $x$  座標，第二個數字為  $y$  座標。 $x$  座標與  $y$  座標之間以一個空白字元隔開。所有點座標之  $x$  座標與  $y$  座標皆為整數。

### 輸出：

依讀入的順序逐一輸出讀入之點座標，點座標的  $x$  座標與  $y$  座標之間以兩個空白字元隔開。一個點座標如果被已經輸出的某個點座標支配，則不再輸出。以下列的輸入範例為例，第四列的座標 $(1, -2)$ 已經被第二列的座標 $(1, -2)$ 支配，故無需輸出。第5列的座標 $(2, 4)$ 已被第3列的座標 $(8, 5)$ 支配，亦無須輸出。輸出的點座標與點座標之間以一個換行字元（'\n'）隔開。輸出最後一個點座標之後輸出一個換行字元（'\n'）做為結束。

### 輸入範例：

```
5
1 -2
8 5
1 -2
2 4
3 6
```

### 輸出範例：

```
1 -2
8 5
3 6
```



## QA\_7\_4 不被包含的區間集

$x$  與  $y$  為實數， $x \leq y$ 。區間  $[x, y]$  為大於等於  $x$  且小於等於  $y$  的所有實數所形成的集合。換言之， $[x, y] = \{z \mid z \text{ 為實數}, x \leq z \leq y\}$ 。我們稱  $x$  與  $y$  分別為區間  $[x, y]$  之左端點與右端點。

區間  $[a, b]$  包含區間  $[x, y]$  若且為若  $a \leq x$  且  $y \leq b$ 。

例如  $[-2, 1]$  包含  $[-1, 1]$ ； $[1, 5]$  包含  $[2, 4]$ ，但是  $[1, 5]$  不包含  $[2, 6]$ ； $[2, 6]$  也不包含  $[1, 5]$ 。一個沒有兩個相等的區間之區間集合  $I$  中不為同集合中另一個區間包含的區間所形成的子集合為  $I$  的極大區間集。寫一個程式，輸入一個區間集合  $I$  輸出其極大區間集。

### 輸入：

輸入共  $n + 1$  列。第一列為區間數目  $n$ ， $0 < n \leq 100000$ 。第二列起每列為一個區間。每列的第一個數字為區間的左端點，第二個數字為區間的右端點。左端點與右端點之間以空白字元隔開。所有左端點與右端點皆為整數。任兩區間皆不相等。

### 輸出：

依讀入的順序逐一輸出每列之區間，一個區間如果被輸入的的  $n$  個區間中另一個區間包含，則不輸出。輸出的區間與區間之間以一個換行字元（'\n'）隔開。輸出最後一個區間之後以一個換行字元（'\n'）結束。

### 輸入範例：

```
5
1 1
2 8
-2 1
2 4
3 6
```

### 輸出範例：

```
2 8
-2 1
```

## QA\_8\_1 合併遞增陣列

令  $Ary[]$  為一個長度為  $n$  的 C 語言陣列。如果對任一  $i$ ， $0 \leq i < n-1$ ，滿足  $Ary[i] \leq Ary[i+1]$ ，則稱其為遞增陣列，例如  $Ary[] = \{-2385\ 158\ 1234\ 1561\ 2233\ 4569\}$  為遞增陣列。將兩個遞增陣列合併成一個遞增陣列是合併排序法 (merge sort) 的核心程序。請寫一個程式輸入兩個遞增整數數列，將其合併成一個遞增整數數列。

### 輸入：

三列：

第一列為正整數  $n$  與  $m$ ， $0 < n, m \leq 100000$ 。

第二列為遞增整數陣列  $A[]$  的  $n$  個整數，

第三列為遞增整數陣列  $B[]$  的  $m$  個整數。

每一列的數與數之間皆以空白字元隔開。

### 輸出：

將輸入的遞增整數陣列  $A[]$  與  $B[]$  合併成一個遞增整數陣列後依序輸出。數與數之間以兩個空白字元隔開。輸出最後一個數之後接著輸出一個換行字元('\n')。

### 輸入範例：

```
9 6
-235 18 24 56 56 56 156 223 459
15 30 56 56 500 500
```

### 輸出範例：

```
-235 15 18 24 30 56 56 56 56 56 156 223 459 500 500
```

## QA\_8\_2 字典順序(lexicographic order)比較

英文字典的英文單字為了方便檢索，通常依字典順序排序。令  $A = a_1a_2\dots a_n$  為一個長度為  $n$  的字串(string)， $B = b_1b_2\dots b_m$  為一個長度為  $m$  的字串。其中  $a_i, 1 \leq i \leq n$ ，和  $b_j, 1 \leq j \leq m$ ，皆為英文字母。定義字典順序  $A < B$  若且唯若存在  $k$  滿足下列條件：

- (1)  $1 \leq k \leq \min(n, m)$ 。
- (2) 所有從 1 到  $k$  的  $i$  值， $a_i = b_i$ 。
- (3)  $k = n < m$  或者  $(k < \min(n, m)$  且  $a_{k+1} < b_{k+1})$ 。

假如  $n = m$  而且對所有從 1 到  $n$  的  $i$  值， $a_i = b_i$ ，則  $A = B$ 。例如 “Ton” > “Tom”，“jack” = “jack”，“name” < “orange”，“keep” < “keeping”。

### 輸入：

$n + 1$  列：

第一列為一個正整數  $n, 0 < n \leq 100000$ 。

第二列到第  $n + 1$  列共  $n$  列，每列有兩個英文單字 A 與 B，單字與單字之間以一個空白字元隔開。每個單字不超過 31 個字母。

### 輸出：

依序針對第二列到第  $n + 1$  列的每一列之英文單字 A 與 B，輸出  $A < B$  或  $A = B$  或者  $A > B$ 。

### 輸入範例：

```
6
ton Tom
keep keeping
lexical lexicographic
weather where
mother mom
tiger tiger
```

### 輸出範例：

```
ton > Tom
keep < keeping
lexical < lexicographic
weather < where
mother > mom
tiger = tiger
```

## QA\_8\_3 排序函式測試

排序是計算機科學一個很重要的運算。很多文獻探討排序演算法。資料結構或演算法入門教科書常以泡沫排序(bubble sort)、插入排序(insertion sort)、選擇排序(selection sort)等方法示範排序程式。其他經典排序演算法還有合併排序(merge sort)與堆積排序(heap sort)。快速排序(quick sort)則為公認最實用的演算法。還有一些排序演算法各有其特別適用的場合，不再贅述。本練習題旨在提供排序測試資料，提供測試各種排序函式之用。

### 輸入：

$n + 1$  列：

第一列為正整數  $n$ ， $0 < n \leq 100000$ 。

第二列起每列為  $m + 1$  個整數， $m$  為該列的第一個數， $0 < m \leq 1000000$ 。每列的數與數之間，以空白字元隔開。

### 輸出：

$n$  列：

依序針對輸入第二列起每列第一個數之後的  $m$  個數 ( $m$  為該列的第一個數) 依由小到大的順序輸出。每列輸出的數與數之間，以兩個空白字元隔開，每列輸出最後一個數字之後，以一個換行字元 ('\n') 結束。

### 輸入範例：

2

6 4569 2233 1234 -2385 158 156

4 1561 -2385 300 4596

### 輸出範例：

-2385 156 158 1234 2233 4569

-2385 300 1561 4596

## QA\_8\_4 第 $k$ 大的數

請寫個程式從  $n$  個數之中找出第  $k$  大的數。第  $k$  大的數之定義為將數依由大至小的順序排列，排在第  $k$  位者。第 1 大的數為所有數中最大者，第  $n$  大的數則為所有數中最小者。假如數並不是兩兩不相等，凡與排在第  $k$  位者相等的數，皆可稱為第  $k$  大的數。

### 輸入：

兩列。

第一列為正整數  $n$  與  $k$ ， $0 < k \leq n$ ， $0 < n \leq 100000$ 。

第二列為  $n$  個整數。

每列的數與數之間以空白字元隔開。

### 輸出：

輸出輸入的第二列的  $n$  個整數之第  $k$  大的數。如果不唯一，只需輸出其中之一。請注意，第 1 大的數表示為所有數中最大者，第  $n$  大的數則為  $n$  個數中最小者。

### 輸入範例：

6 3

2233 1234 -2385 1235 1230 1561

### 輸出範例：

1235

## QA\_8\_5 等價陣列之判斷

兩個 C 語言陣列 A[] 與 B[] 如果滿足下列兩個條件，稱這兩個陣列**等價**：

1. 它們的長度相等。
2. 令  $n$  為它們的長度。存在一個從  $\{0, 1, 2, \dots, n-1\}$  對應到  $\{0, 1, 2, \dots, n-1\}$  的一對一且映成的函數  $f$ ，使得任一  $k, 0 \leq k < n$ ，皆滿足  $A[k] == B[f(k)]$ 。

假設  $A[] = \{4569\ 1234\ 1234\ -2385\}$ ， $B[] = \{1234\ -2385\ 4569\ 1234\}$ 。存在函數  $f$  其函數對映如下：

0 ---→ 2  
1 ---→ 0  
2 ---→ 3  
3 ---→ 1

使得

$A[0] == B[f(0)] == B[2] == 4596$   
 $A[1] == B[f(1)] == B[0] == 1234$   
 $A[2] == B[f(2)] == B[3] == 1234$   
 $A[3] == B[f(3)] == B[1] == -2385$

因此  $A[] = \{4569\ 1234\ 1234\ -2385\}$  與  $B[] = \{1234\ -2385\ 4569\ 1234\}$  等價。  
請寫一個程式，輸入 A[] 與 B[] 兩陣列之元素，判斷它們是否等價。

### 輸入：

三列：

第一列為陣列長度  $n$ ， $0 < n \leq 100000$ 。

第二列與第三列分別為陣列 A[] 與 B[] 之元素，它們的元素個數皆為  $n$ 。每列的數與數之間以空白字元隔開。

### 輸出：

如果輸入的陣列 A[] 與 B[] 為等價則輸出 “Yes!”；否則輸出 “No!”。之後輸出一個換行字元(‘\n’)做為結束。

輸入範例一：

6

4569 2233 1234 -2385 2233 156

2233 -2385 1234 2233 156 4569

輸出範例一：

Yes!

輸入範例二：

6

4569 2233 1234 -2385 2233 156

156 -2385 1234 2233 2233 5496

輸出範例二：

No!

## QA\_9\_1 陣列中和為零的段落

已知  $A[]$  為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。  $A[]$  的任兩元素的值不相等。寫一程式判斷是否存在  $h$  與  $k$ ， $0 \leq h \leq k < n$ ，滿足

$$A[h] + A[h + 1] + \dots + A[k] = 0。$$

### 輸入：

兩列：

第一列為陣列  $A[]$  的元素個數  $n$ ， $0 < n \leq 100000$ 。

第二列為陣列  $A[]$  的  $n$  個元素，數與數之間以空白字元隔開。

### 輸出：

如果存在  $h$  與  $k$ ， $0 \leq h \leq k < n$ ，滿足

$$A[h] + A[h + 1] + \dots + A[k] = 0，$$

則輸出  $h$  與  $k$ ；如果存在多組  $h$  與  $k$ ，輸出  $h$  最大之那一組， $h$  與  $k$  之間以兩個空白字元隔開；若不存在，則輸出 “It does not exist!”。以下列輸入範例之陣列為例： $A[] = \{-3, -7, -6, -5, -4, -8, 33, 102, 98, -200, 99\}$ 。觀察得之，共有  $A[h=0] + A[1] + \dots + A[k=6] = (-3) + (-7) + (-6) + (-5) + (-4) + (-8) + 33 = 0$ ，以及  $A[h=7] + A[8] + A[k=9] = 102 + 98 + (-200) = 0$ 。依題意要求，如果存在多組  $h$  與  $k$ ，輸出  $h$  最大之那一組，故輸出 “7 9”。

### 輸入範例：

11

-3 -7 -6 -5 -4 -8 33 102 98 -200 99

### 輸出範例：

7 9



## QA\_9\_2 陣列等值首尾和位置

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。`intAry[]` 的每個元素都大於 0。寫一程式判斷是否存在  $i$ ， $0 \leq i < n - 1$ ，滿足  $X_i = Y_i$  其中  $X_i = \text{intAry}[0] + \text{intAry}[1] + \dots + \text{intAry}[i]$ ，

$Y_i = \text{intAry}[i + 1] + \text{intAry}[i + 2] + \dots + \text{intAry}[n - 1]$ 。

例如，假設  $C = \{1, 2, 3, 7, 6, 5, 4, 8, 36\}$  共 9 個元素，前八個元素的和 36 等於後面一個元素的和，因此輸出 7。但如果  $C = \{1, 2, 3, 7, 6, 5, 4, 8, 29, 80\}$  共 10 個元素，則不存在  $i$  滿足  $X_i = Y_i$ 。因此輸出 It does not exist!

### 輸入：

輸入共兩列。第一列為 C 語言整數陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。第二列為陣列 `intAry[]` 的  $n$  個元素，元素與元素之間以空白字元隔開。

### 輸出：

如果存在  $i$ ， $0 \leq i < n - 1$ ，滿足  $X_i = Y_i$  其中  $X_i = \text{intAry}[0] + \text{intAry}[1] + \dots + \text{intAry}[i]$   $Y_i = \text{intAry}[i + 1] + \text{intAry}[i + 2] + \dots + \text{intAry}[n - 1]$ ；則輸出  $i$ 。若不存在，則輸出 “It does not exist!\n”

### 輸入範例一：

```
9
1 2 3 7 6 5 4 8 36
```

### 輸出範例一：

```
7
```

### 輸入範例二：

```
9
1 2 3 7 6 5 4 8 39
```

### 輸出範例二：

```
It does not exist!
```

## QA\_9\_3 整數陣列中差異最小的兩數

在一堆數字之中尋找差異最小的兩數是程式設計常見的程序。兩數的差異是指該兩數的差的絕對值。

### 輸入：

$n + 1$  列：

第一列為以空白字元隔開的兩個正整數  $n$  與  $m$ ， $0 < n$ ， $m \leq 100000$ 。

第二列起每列有  $m$  個數，數與數之間以空白字元隔開。

### 輸出：

針對第一列之後的每一列輸入，輸出一列內含該列輸入的  $m$  個數字中差異最小的兩數，比較小的先輸出，數與數之間以兩個空白字元隔開。如果有兩對以上數字差異為最小，輸出每對數字比較小的數字中最大者。例如輸入範例之最後一列，應輸出 1158 1300 而非 158 300。

### 輸入範例：

```
5 5
1234 -2385 158 1561 2233
8889 2345 157 300 8889
4521 300 157 4523 8889
157 257 300 4521 8889
300 1158 158 1834 1300
```

### 輸出範例：

```
1234 1561
8889 8889
4521 4523
257 300
158 300
```

## QA\_9\_4 最長等值子陣列之長度與值

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ ，元素已經由小到大排列。稱 `intAry[]` 陣列中從索引  $i$  到索引  $j$ ， $0 \leq i \leq j < n$ ，連續的一段元素為 `intAry[]` 的子陣列。若 `intAry[]` 的子陣列中的元素均相等，則稱其為 `intAry[]` 的等值子陣列。假設 `intAry[] = {1, 1, 2, 7, 7, 7, 7, 8, 36, 36}`，則  $\{1, 1\}$ 、 $\{2\}$ 、 $\{7, 7\}$ 、 $\{7, 7, 7, 7\}$ 、 $\{8\}$ 、 $\{36, 36\}$  等均為 `intAry[]` 之等值子陣列。等值子陣列內元素之值為等值子陣列之值。等值子陣列之長度為等值子陣列內之元素個數。根據定義，`intAry[]` 最長之等值子陣列為  $\{7, 7, 7, 7\}$  之值為 7，其長度為 4。

### 輸入：

輸入共兩列。第一列為 C 語言整數陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。第二列為陣列 `intAry[]` 的  $n$  個元素，元素與元素之間以空白字元隔開。  
**陣列 `intAry[]` 的  $n$  個元素已經依由小到大的順序排列。**

### 輸出：

`intAry[]` 最長之等值子陣列之長度以及其值，兩數之間以兩個空白字元隔開。假如有兩個以上的等值子陣列長度皆等於最長之等值子陣列之長度，輸出其中值最大者之值。

### 輸入範例：

```
10
21 21 22 37 37 37 37 48 56 56
```

### 輸出範例：

```
4 37
```

## QA\_9\_5 最長等值子陣列之位置

已知 `intAry[]` 為  $n$  個元素的 C 語言整數陣列， $0 < n \leq 100000$ 。稱 `intAry[]` 陣列中從索引  $i$  到索引  $j$ ， $0 \leq i \leq j < n$ ，連續的一段元素為 `intAry[]` 的子陣列，用 `intAry[i, j]` 來表示。稱  $(i, j)$  為 `intAry[i, j]` 在 `intAry[]` 中的位置。若 `intAry[i, j]` 中的元素均相等，則稱其為 `intAry[]` 的等值子陣列。假設 `intAry[] = {1, 1, 36, 7, 7, 7, 7, 5, 36, 36}`，則  $\{1, 1\}$ 、 $\{36\}$ 、 $\{7, 7\}$ 、 $\{7, 7, 7, 7\}$ 、 $\{5\}$ 、 $\{36, 36\}$  等均為 `intAry[]` 之等值子陣列。等值子陣列內元素之值為等值子陣列之值。等值子陣列之長度為等值子陣列內之元素個數。根據定義，`intAry[]` 最長之等值子陣列為 `intAry[3, 6] = {7, 7, 7, 7}`。

### 輸入：

兩列：

第一列為 C 語言整數陣列 `intAry[]` 的元素個數  $n$ ， $0 < n \leq 100000$ 。

第二列為陣列 `intAry[]` 的  $n$  個元素，元素與元素之間以空白字元隔開。

### 輸出：

`intAry[]` 最長等值子陣列之位置  $(i, j)$ 。最後輸出一個換行字元（'\n'）。假如有兩個最長等值子陣列，輸出  $i$  值最大者之位置。例如輸入範例之陣列有兩個長度一樣的最長子陣列，其位置分別為  $(3, 6)$  與  $(8, 11)$ 。 $i$  值較大者之位置為  $(8, 11)$ ，故輸出  $(8, 11)$ 。

### 輸入範例：

12

21 21 48 37 37 37 37 22 56 56 56 56

### 輸出範例：

(8, 11)

## QA\_10\_1 分屬不同的陣列的兩元素之和

令  $A[]$  與  $B[]$  為 C 語言整數陣列，各有  $n$  與  $m$  個元素。寫一個程式，逐一讀入  $s$  個整數  $S$ ，判斷是否存在  $i$  和  $j$ ， $0 \leq i < n$ ， $0 \leq j < m$ ，滿足  $A[i] + B[j] == S$ 。

### 輸入：

四列：

第一列為三個整數  $n$ 、 $m$  與  $s$ ， $0 < n \leq 100000$ ， $0 < m \leq 100000$ ， $0 < s \leq 100000$ 。  
 $n$  與  $m$  分別為陣列  $A[]$  與  $B[]$  元素的個數。

第二列依序為陣列  $A[]$  的  $n$  個元素。

第三列依序為陣列  $B[]$  的  $m$  個整數元素。

第四列為  $s$  個整數  $S$ 。

每列的數與數之間皆以空白字元隔開。

### 輸出：

$s$  列：

對輸入的第四列之每一個整數  $S$ ，若存在  $i$  和  $j$ ， $0 \leq i < n$ ， $0 \leq j < m$ ，滿足  $A[i] + B[j] == S$ ，輸出滿足條件的最小  $i$  值，以及與該最小  $i$  值配合滿足條件的最大  $j$  值，兩數之間以兩個空白字元隔開。若不存在，則輸出 “It does not exist!”

### 輸入範例：

```
9 6 2
-10 99 11 3 11 17 19 4 22
78 89 -23 78 6 89
100 200
```

### 輸出範例：

```
2 5
It does not exist!
```

## QA\_10\_2 分屬不同的遞增陣列的兩元素之和

令  $A[]$  為一個長度為  $n$  的 C 語言陣列。如果對任一  $i$ ,  $0 \leq i < n-1$ , 滿足  $A[i] \leq A[i+1]$ , 則稱其為遞增陣列, 例如

$A[] = \{-2385, 158, 1234, 1234, 1561, 2233, 4569\}$  為遞增陣列。

已知  $A[]$  與  $B[]$  皆為 C 語言遞增的整數陣列, 各有  $n$  與  $m$  個元素。寫一個程式, 逐一讀入  $s$  個整數  $S$ , 判斷是否存在  $i$  和  $j$ ,  $0 \leq i < n, 0 \leq j < m$ , 滿足  $A[i] + B[j] == S$ 。

### 輸入：

四列：

第一列為三個整數  $n$ 、 $m$  與  $s$ ,  $0 < n \leq 100000, 0 < m \leq 100000, 0 < s \leq 100000$ 。

$n$  與  $m$  分別為遞增的整數陣列  $A[]$  與  $B[]$  元素的個數。

第二列依序為陣列  $A[]$  的  $n$  個元素。

第三列依序為陣列  $B[]$  的  $m$  個整數元素。

第四列為  $s$  個整數。每列的數與數之間皆以空白字元隔開。

### 輸出：

$s$  列：

對輸入的第四列之每一個整數  $S$ , 若存在  $i$  和  $j$ ,  $0 \leq i < n, 0 \leq j < m$ , 滿足  $A[i] + B[j] == S$ , 輸出滿足條件的最小  $i$  值, 以及與該最小  $i$  值配合滿足條件的最大  $j$  值, 兩數之間以兩個空白字元隔開。若不存在, 則輸出 “It does not exist!”

### 輸入範例：

9 5 2

-10 3 4 5 11 11 17 19 22

-23 6 78 89 89

100 200

### 輸出範例：

4 4

It does not exist!

## QA\_10\_3 兩陣列的等值對

已知A[] 與B[] 為兩個 C 語言整數陣列，各有  $n$  與  $m$  個元素。A[] 中元素兩兩不相等，B[] 中元素亦然。若存在  $i$  和  $j$ ， $0 \leq i < n$ ， $0 \leq j < m$ ，滿足  $A[i] = B[j]$ ，我們稱  $(i, j)$  為A[] 與B[] 的等值對。兩陣列的等值對不一定存在；若存在，不一定唯一。

### 輸入：

三列。

第一列為兩個整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 100000$ 。 $n$  與  $m$  分別為 C 語言整數陣列 A[] 與 B[] 的元素個數。

第二列依序為陣列 A[] 的  $n$  個整數，A[] 中整數兩兩不相等。

第三列依序為陣列 B[] 的  $m$  個整數，B[] 中整數兩兩不相等。

每列的數與數之間以空白字元隔開。

### 輸出：

一列。依下列規則輸出之後，輸出一個換行字元（'\n'）：

若陣列 A[] 與 B[] 的存在等值對，依**字典順序由大到小**輸出陣列 A[] 與 B[] 的所有等值對。等值對與等值對之間以兩個空白字元隔開。若不存在等值對，則輸出 “It does not exist!”。

### 輸入範例一：

```
9 5
-10 11 22 33 34 56 89 99 101
81 22 78 99 89
```

### 輸出範例一：

```
(7, 3) (6, 4) (2, 1)
```

### 輸入範例二：

```
9 5
-10 11 221 33 34 56 889 99 101
81 22 78 999 89
```

### 輸出範例二：

```
It does not exist!
```

## QA\_10\_4 兩陣列的距離

已知 A[] 與 B[] 皆為 C 語言整數陣列，各有  $n$  與  $m$  個元素。定義 A[] 與 B[] 的距離為分屬 A[] 與 B[] 的兩數的差的絕對值的最小值。假如 A[] 與 B[] 有共同的元素，根據定義兩者之距離為 0。例如

A[] = {-10, 3, 4, 5, 11, 22, 11, 17, 19}

B[] = {-23, 6, 89, 78, 89}

則 A[] 與 B[] 之間的距離為 1。這是因為分屬 A[] 與 B[] 的兩數的差的絕對值的最小值為  $|A[3] - B[1]| = |5 - 6| = 1$ 。

寫一程式求兩陣列之距離。

### 輸入：

三列。

第一列為兩個整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 100000$ 。 $n$  與  $m$  分別為 C 語言整數陣列 A[] 與 B[] 元素的個數。

第二列依序為陣列 A[] 的  $n$  個整數。

第三列依序為陣列 B[] 的  $m$  個整數。

同一列的數與數之間以空白字元隔開。

### 輸出：

輸出陣列 A[] 與陣列 B[] 的距離後，輸出一個換行字元('\n')。

### 輸入範例：

9 5

-10 3 4 5 11 22 11 17 19

-23 6 89 78 89

### 輸出範例：

1



## QA\_10\_5 求兩遞增陣列的等值對

令  $A[]$  為一個長度為  $n$  的 C 語言陣列。如果對任一  $i$ ,  $0 \leq i < n-1$ , 滿足  $A[i] \leq A[i+1]$ , 則稱其為遞增陣列, 例如  $A[] = \{-2385\ 158\ 1234\ 1561\ 2233\ 4569\}$  為遞增陣列。反之, 如果對任一  $i$ ,  $0 \leq i < n-1$ , 滿足  $A[i] \geq A[i+1]$ , 則稱其為遞減陣列, 例如  $A[] = \{5881\ 4521\ 2345\ 2345\ 300\ 157\}$  為遞減陣列。已知  $A[]$  與  $B[]$  為兩個 C 語言遞增整數陣列, 各有  $n$  與  $m$  個元素。 $A[]$  中元素兩兩不相等,  $B[]$  中元素亦然。若存在  $i$  和  $j$ ,  $0 \leq i < n, 0 \leq j < m$ , 滿足  $A[i] = B[j]$ , 我們稱  $(i, j)$  為  $A[]$  與  $B[]$  的一等值對。

### 輸入：

三列。

第一列為兩個整數  $n$  與  $m$ ,  $0 < n \leq 100000, 0 < m \leq 100000$ 。

第二列依序為遞增整數陣列  $A[]$  的  $n$  個整數,  $A[]$  中整數兩兩不相等。

第三列依序為遞增整數陣列  $B[]$  的  $m$  個整數,  $B[]$  中整數兩兩不相等。

每列的數與數之間以空白字元隔開。

### 輸出：

若陣列  $A[]$  與  $B[]$  的存在等值對, 依字典順序由大到小輸出陣列  $A[]$  與  $B[]$  的所有等值對。等值對與等值對之間以兩個空白字元隔開。若不存在等值對, 則輸出 “It does not exist!”。最後再輸出一個換行字元(“\n”)。

### 輸入範例一：

```
10 6
-10 11 22 33 34 56 89 89 99 101
22 78 81 89 89 99
```

### 輸出範例一：

```
(6, 3) (6, 4) (7, 3) (7, 4) (8, 5)
```

### 輸入範例二：

```
10 6
-10 11 22 33 34 56 189 189 199 201
20 78 81 89 89 99
```

### 輸出範例二：

```
It does not exist!
```

## QA\_10\_6 求兩遞增陣列的距離

令  $Ary[]$  為一個長度為  $n$  的 C 語言陣列。如果對任一  $i$ ,  $0 \leq i < n - 1$ , 滿足  $Ary[i] \leq Ary[i + 1]$ , 則稱其為遞增陣列, 例如  $Ary[] = \{-2385\ 158\ 1234\ 1561\ 2233\ 4569\}$  為遞增陣列。已知  $A[]$  與  $B[]$  皆為 C 語言遞增整數陣列, 各有  $n$  與  $m$  個元素。定義  $A[]$  與  $B[]$  的距離為分屬  $A[]$  與  $B[]$  的兩數的差之絕對值的最小值。假如  $A[]$  與  $B[]$  有共同的元素, 根據定義兩者之距離為 0。例如

$A[] = \{-10, 3, 4, 5, 11, 11, 17, 19, 22\}$

$B[] = \{-23, 6, 78, 89, 89\}$

則  $A[]$  與  $B[]$  之間的距離為 1。這是因為分屬  $A[]$  與  $B[]$  的兩數的差的絕對值的最小值為  $|A[3] - B[1]| = |5 - 6| = 1$ 。

寫一程式, 輸入兩遞增陣列, 輸出他們的距離。

### 輸入：

輸入共有三列。第一列為兩個整數  $n$  與  $m$ ,  $0 < n \leq 100000$ ,  $0 < m \leq 100000$ 。 $n$  與  $m$  分別為 C 語言陣列  $A[]$  與  $B[]$  元素的個數。第二列依序為陣列  $A[]$  的  $n$  個整數, 第三列依序為陣列  $B[]$  的  $m$  個整數。同一列的數與數之間以空白字元隔開。陣列  $A[]$  與  $B[]$  皆為遞增陣列。

### 輸出：

輸出如下文字於一列：

The distance of  $A[]$  and  $B[]$  is  $k$ .

其中  $k$  為陣列  $A[]$  與陣列  $B[]$  的距離。

### 輸入範例：

9 5

-10 3 4 5 11 11 17 19 22

-23 6 78 89 89

### 輸出範例：

The distance of  $A[]$  and  $B[]$  is 1.

## QA\_10\_7 集合的交集

已知 A 與 B 為兩個整數集合，各有  $n$  與  $m$  個元素，無論 A 或 B 集合的任兩元素皆不相等。集合 A 與 B 的交集為同時出現在集合 A 與 B 中的元素所形成的集合。例如，同時出現在集合  $A = \{-10, 3, 4, 5, 11, 13, 17, 19, 22\}$  與  $B = \{-23, 6, 17, 22, 89\}$  中的元素為 17 與 22，故 A 與 B 的交集為  $\{17, 22\}$ 。

### 輸入：

輸入共  $3T + 1$  列。第一列為測試資料的組數  $T$ 。第二列起每三列為一組測試資料。每組測試資料的第一列為兩個整數  $n$  與  $m$ ， $0 < n \leq 100000$ ， $0 < m \leq 100000$ 。 $n$  與  $m$  分別為集合 A 與 集合 B 元素的個數。每組測試資料的第二列依序為集合 A 的  $n$  個整數，第三列依序為集合 B 的  $m$  個整數。每一列的數與數之間皆以空白字元隔開。每組測試資料的集合 A 與 集合 B 的元素皆依由小到大的順序排列。

### 輸出：

針對每組測試資料輸出兩列，第一列為 A 與 B 的交集的元素個數；第二列以由小到大的順序依序輸出 A 與 B 的交集之所有元素。元素與元素之間以兩個空白字元隔開。最後一個元素之後以一個換行字元 ('\n') 結束。假如交集為空集合，意即沒有任何元素同時出現在 A 與 B 之中，則輸出一列

“The intersection of A and B is empty!”

### 輸入範例：

```
2
9 5
-10 3 4 5 11 13 17 19 22
-23 6 17 22 89
9 5
-10 3 4 5 11 14 18 19 22
-23 6 17 25 89
```

### 輸出範例：

```
2
17 22
The intersection of A and B is empty!
```

## QA\_10\_8 尋找三陣列的共同元素

已知  $A[]$ ， $B[]$  與  $C[]$  為三個元素皆已經由小到大排列妥當的整數陣列，各有  $n$ 、 $m$  與  $p$  個元素。寫一個程式，判斷是否存在  $i$ ， $j$  和  $k$ ， $0 \leq i < n$ ， $0 \leq j < m$ ， $0 \leq k < p$  滿足  $A[i] = B[j] = C[k]$ 。

### 輸入：

四列：

第一列為三個整數  $n$ ， $m$  與  $p$ ， $0 < n \leq 100000$ ， $0 < m \leq 100000$ ， $0 < p \leq 100000$ 。

第二列依序為整數陣列  $A[]$  的  $n$  個元素。

第三列依序為整數陣列  $B[]$  的  $m$  個元素。

第四列依序為個整陣列  $C[]$  的  $p$  數元素。

三個陣列的元素皆已經由小到大排列。每列的數與數之間以空白字元隔開。

### 輸出：

若存在一數同時為陣列  $A[]$ 、 $B[]$  與  $C[]$  的元素，輸出同時為三個陣列的元素之最小值。若不存在，則輸出 “It does not exist!”。最後再輸出一個換行字元(‘\n’)。

### 輸入範例一：

```
9 5 2
-10 11 22 33 34 56 89 99 101
6 78 81 89 99
89 99
```

### 輸出範例一：

```
89
```

### 輸入範例二：

```
9 5 2
-10 11 22 33 34 56 89 99 101
6 78 81 89 99
88 98
```

### 輸出範例二：

```
It does not exist!
```