

-
1. 請寫一個程式，從資料檔(a.txt)讀入一個正整數，把它的所有質因數找出來。例如：如果輸入是 72，因為 $72=2^3 \times 3^2$ ，於是質因數就有 2 與 3，如果輸入是 181944，因為 $181944=19^2 \times 7 \times 3^2 \times 2^3$ ，於是質因數就有 19，7，3 與 2。為方便起見， $19^2 \times 7 \times 3^2 \times 2^3$ ，使用 19(2)7(1)3(2)2(3) 作輸出。也就是說， a^b 輸出為 a(b)。

範例輸入：

181944

範例輸出：

181944

Ans=19(2)7(1)3(2)2(3)

2. 對於一個字串若起始字元為 A，最後字元為 L，且其間不含任何 A 或 L 字元則稱 AL 字串。

例： A□PENCIL 或 A□XXBBHCL (四方形表示空白字元)

請設計一程式能從資料檔 b.txt 讀入一個字串 (不超過 200 個字元)，然後消去所有可能的 AL 字串，使得消去後之輸出不包含 AL 字串。

範例輸入：

This□is□a□ball

範例輸出：

This□is□a□ball

Ans=This□is□

3. 從檔案(c.txt)讀入一算術運算式，計算並列印其結果。

(1)運算式包含+*/四則算術運算

(2)輸入數字可能包含小數點

(3)不分+*/一律由左而右依序計算

(4)運算式的最後結果依四捨五入後取整數值

以下幾個範例可供參考測試之：

範例輸入一：

15*3/8*10/3=

範例輸出一：

15*3/8*10/3=19

範例輸入二：

6.53*5+3-8=

範例輸出二：

6.53*5+3-8=28

4. 請從資料檔 d.txt 讀入好幾行字串資料，字串資料是由字元 ('a'~'z', 'A'~'Z')、數字 ('0'~'9') 以及空白字元所構成的，接著將字串中這些由字元或者空白隔開的數字部分進行加總 ($54+256+72+125+56+22+66=751$)，並且將原字串以及結果輸出。

範例輸入：

Cyut54is a356good 72school

Ne125ver 56give22 66up

計算方法： $54+356+72+125+56+22+66=751$

範例輸出：

Cyut54is a356good 72school

Ne125ver 56give22 66up

Ans=751

5. 試寫一程式，從檔案 **e.txt** 讀入一串整數，計算出此串整數中某部分連續整數和為最大值，並由螢幕輸出。

範例輸入：

1 2 -6 3 -2 4 -1 3 2 -4

計算方法：最大值是 $3+(-2)+4+(-1)+3+2=9$

範例輸出：

1 2 -6 3 -2 4 -1 3 2 -4

Ans=9

6. 試寫一程式，從檔案 **(c.txt)** 讀入一串正整數(最多 10 個)，試找出此串正整數之最小公倍數。例如：

範例輸入：

12 16 36 128

範例輸出：

12 16 36 128

Ans=1152

7. 請撰寫一程式，此程式從 **b.txt** 中讀取一個正的 數字(含小數)後將此十進位數字轉換成二進位數字，轉換後的數字請取至小數點第三位(小數點第三位以後的數字直接捨去)。輸出格式為：轉換後的整數+轉換後的小數部分

範例輸入：

21.33

說明：

21.33 轉換成二進位數字為 10101.010101...

取小數點後三位後的數字為 10101.010

因此輸出結果為 10101+010

範例輸出：

21.33

Ans=10101+010

8. 內插法：為了擴大數值的數量，內插是常見的作法，通常需要增加數值的位置，就以該位置前一個數值與該位置的數值，計算平均並取四捨五入後插入，舉例來說，假設一串數值如下：

3 2 5 7 8 1

內插間隔為 2，表示每 2 個數值之後插入 1 個數值，所以插入的過程如下：

第 1 個插入位置為 3，所以插入位置數值 5 與前一個位置數值 2，取平均後得到 4

第 2 個插入位置為 5，所以插入位置數值 8 與前一個位置數值 7，取平均後得到 8

第 3 個插入位置為 7，沒有數值，所以內插動作結束。

插入後的結果為：

3 2 4 5 7 8 8 1

輸入檔說明：

輸入檔 **e.txt** 中包含兩行文字，第一行為一連串整數，個數不超過 100 個，每個數字間以空白間隔。第二行是內插的間隔

範例輸入：

3 2 5 7 8 1
1

範例輸出：

3 2 5 7 8 1
1

Ans=3 3 2 4 5 6 7 8 8 5 1

-
9. 恐怖份子在一條巷子中放入了好幾顆炸彈，炸彈有分小炸彈與大炸彈，請計算這條巷子所有炸彈引爆後，還有那些地方是安全的。這裡假設巷子以一連串的 0 來表示，小 x 表示小炸彈，爆炸範圍是 1 (會炸到左右各一個 0)，大 X 是大炸彈，爆炸範圍是 2 (會炸到左右各 2 個 0)，請從 **d.txt** 中讀入一串由 0, x, X 組成的字串 (最長 80 個字元)，接著輸出炸彈爆炸後還剩下幾個 0。

範例輸入一：

00x00000X0xX0000x

上面的炸彈爆炸後的結果為：

0---00-----0--

剩下 4 個 0

範例輸出一：

00x00000X0xX0000x

Ans=4

-
10. 分別自檔案 **e.txt** 讀入一串由 O[]{} 所組成之字串，判斷字串是否合法，一個 O[]{} 必須對稱存在，且對應的括號間不能插入單獨的括號。

範例輸入一：

({[()] () })

範例輸出一：

({[()] () })

Ans=CORRECT

範例輸入二：

() {[()] ()

範例輸出二：

() {[()] ()

Ans=ERROR

-
11. 中國餘數定理最早記載於第一世紀之孫子算經中，其原題目為：今有物不知其數，三三數之剩二，五五數之剩三，七七數之剩二，問物幾何？

其計算過程如下：

1. 首先計算所有質數的乘積 -> $N=3*5*7=105$

2. 計算 N 除以每個質數的商 -> $105/3=35$, $105/5=21$, $105/7=15$

3. 計算這個商的倍數值，使得對原質數取餘數的結果為 1

$35*y_1 \bmod 3=1$ 得到 $y_1=2$

$21 * y_2 \bmod 5 = 1$ 得到 $y_2 = 1$

$15 * y_3 \bmod 7 = 1$ 得到 $y_3 = 1$

4. 答案 x 等於每個質數的商、倍數值、餘數三者乘積的總和 \bmod 原來質數的乘積

$$x = 35 * 2 * 2 + 21 * 1 * 3 + 15 * 1 * 2 \bmod 105 = 23$$

請撰寫一個程式從資料檔 **e.txt** 依序讀入幾個質數（不一定只有三個質數）與對應的餘數，然後輸出解答。

範例輸入：

3 2 5 3 7 2

範例輸出：

3 2 5 3 7 2

Ans=23

-
12. 雜湊(hashing)是用來計算一個鍵值(key value)對應之索引(index)的方法，而「折疊法」是雜湊函數的方法之一，假設有一個鍵值為 5120969190600791，而折疊的長度為 3，折疊的計算過程如下，首先鍵值先依照折疊長度加以切割，由於鍵值有 16 個數字，每 3 個數字進行切割的話，可以切出 6 段，如下所示：

512 096 919 060 079 1

接著這些數值進行加總就可以得到所需要的索引 1667，如下所示：

$$512 + 096 + 919 + 060 + 079 + 1 = 1667$$

所以請撰寫一個程式從 **b.txt** 中讀入兩行資料，第一行是鍵值的字串，第二行是折疊的長度，接著輸出索引的結果。

範例輸入：

5120969190600791

3

範例輸出：

5120969190600791

3

Ans=1667

-
13. 異常值(outlier)的移除是資料分析中常常需要進行的步驟，而標準差則是用來判斷異常值的依據，也就是當某個數值未落在平均值 \pm 標準差的範圍內，則判定為異常值，下面的一連串數值($n=11$)為例，分別為 $x_1 \sim x_{11}$ ：

25, 26, 38, 34, 3, 33, 23, 85, 70, 28, 27

其平均值 \bar{x} 為 35.636，標準差 σ 為 22.796，標準差 σ 的公式為 $\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$

所以正常數值範圍為 35.636-22.796 ~ 35.636+22.796

也就是 12.84~58.432

上面這一連串數值中，未落在此範圍的數值為 3, 85, 70，一共有 3 個異常值

請從 **e.txt** 中讀入一連串的整數(可正可負，最多 30 個)，接著，輸出有幾個異常值。

範例輸入一：

25, 26, 38, 34, 3, 33, 23, 85, 70, 28, 27

範例輸出一：

25, 26, 38, 34, 3, 33, 23, 85, 70, 28, 27

Ans=3

14. 身份證號碼的最後一碼為檢查碼，用於確認號碼是否正確，此檢查碼運算規則如下：

1. 英文代號先以下表轉換成數字

A=10 B=11 C=12 D=13 E=14 F=15 G=16 H=17 I=34

J=18 K=19 L=20 M=21 N=22 O=35 P=23 Q=24 R=25

S=26 T=27 U=28 V=29 W=32 X=30 Y=31 Z=33

2. 英文轉成的數字，個位數乘 9 再加上十位數

3. 各數字從右到左依次乘 1, 2, 3, 4, 5, 6, 7, 8

4. 計算上面兩項的總和，計算總和除 10 後之餘數，再用 10 減該餘數，結果就是檢查碼，若餘數為 0，檢查碼就是 0。

舉例來說：身分證號碼是 W100232736，身分證號碼的前 9 個字元為 W10023273

W	1	0	0	2	3	2	7	3
3	2							
*	*	*	*	*	*	*	*	*
1	9	8	7	6	5	4	3	2

$3+18+8+0+0+10+12+6+14+3=74$

$74/10=7\dots4$ (餘數)

$10-4=6$ (檢查碼)

請撰寫一個程式從資料檔 **c.txt** 讀入一個身分證號碼的前 9 個字元，然後輸出檢查碼。

範例輸入：

W10023273

範例輸出：

W10023273

Ans=W100232736

15. 請撰寫一個程式從檔案 **e.txt** 中讀入二個正整數 n 與 k ($n \leq 100000000$)，接著輸出 $0 \sim n$ 的數值中，每個數字總合為 k 的個數。

範例輸入：

100 5

表示要從 $0 \sim 100$ 中找出數字總合為 5 的數值個數，這些數值有：

5

14 (因為 $1 + 4 = 5$)

23 (因為 $2 + 3 = 5$)

32 (因為 $3 + 2 = 5$)

41 (因為 $4 + 1 = 5$)

50 (因為 $5 + 0 = 5$)

範例輸出：

100 5

Ans=6

16. 儲蓄是一種美德，目前常見的儲蓄方式為定時存款，整筆領回，利率採每年複利的方式計算。例如：每年存 100000，存 3 年，若是複利率為 2.25%，3 年後的本利和如下：

第一年初存入 100000，本金=100000

第二年初，前一年本利= $100000 \times (1+2.25\%) = 102250$ ，加上存入的 100000，合計 202250

第三年初，前一年本利= $202250 \times (1+2.25\%) = 206801$ ，加上存入的 100000，合計 306801

第三年尾合計領回的本利和= $306801 \times (1+2.25\%) = 313704$

請寫一個程式從資料檔 **a.txt** 中讀入三行資料，第一行是每年存款的金額，第二行是存款年數，第三行是本利和，接著反算出存款複利率為何（請計算到小數點後兩位）。

範例輸入：

100000

3

313704

範例輸出：

100000

3

313704

Ans=2.25%

17. 中值濾波演算法常用於影像處理，底下使用一串資料來說明演算法的運作：

假設輸入一串正整數如下：

5 16 9 20 3 14 6 8 9 18 32 2

濾波器需要設定一個視窗大小， K (奇數：3, 5, ...)，假設 $K=3$ ，將該串數字從左到右，每 3 個數字排序後，輸出排在中間的數值，針對上面的例子，輸出結果如下：

5 → 字串一開始，個數不足 3 個不處理濾波動作

5 16 → 字串一開始，個數不足 3 個不處理濾波動作

5 16 9 → 排序後 5 9 16，所以輸出 9

16 9 20 → 排序後 9 16 20，所以輸出 16

9 20 3 → 排序後 3 9 20，所以輸出 9

...

上面正整數的輸出結果為：

9 16 9 14 6 8 8 9 18 18

所以，數字中幾個特別大或小的數字就被過濾掉了，如：3 20 32 2

請從資料檔 **e.txt** 中第一行讀入一串正整數，個數 N (最多 100 個)，以及第二行讀入一個 K 值 (奇數 $K \geq 3$, $K \leq N$)，接著輸出濾波結果。

範例輸入：

5 16 9 20 3 14 6 8 9 18 32 2

3

範例輸出：

5 16 9 20 3 14 6 8 9 18 32 2

3

Ans=9 16 9 14 6 8 8 9 18 18

18. 小明設計了三個字串的運算符號，運算方式如下：

~:後面銜接一個數字 N ，將後面的 N 個字元進行反轉，例如：~3abc 得到 cba

@:後面銜接一個數字 N ，表示刪除目前運算結果的起始 N 個字元，

?:後面銜接一個數字 N ，表示刪除目前運算結果的最後 N 個字元

若是沒有任何符號的字串則是依序串接後輸出，請從資料檔 **e.txt** 讀入一個字串 (最長 80 個字元)，該字串由大小寫 a~z 以及上面三種符號與對應數值所組成的情況下，運算

順序也是從左到右，請撰寫程式來輸出運算後的結果。

範例輸入：

The~3UYCTIsNot%3good@3~6eciohc

(上面輸入字串的運算過程如下：)

輸入	目前結果
The	The
~3UYC	TheCYU
TIsNot	TheCYUTIsNot
%3	TheCYUTIs
good	TheCYUTIsGood
@3	CYUTIsGood
~6eciohc	CYUTIsGoodchoice

範例輸出：

The~3UYCTIsNot%3good@3~6eciohc

Ans=CYUTIsGoodchoice

-
19. 部分序列字串指原字串中刪掉一個以上字元且不改變原順序的部份字串。請撰寫程式從資料檔 **d.txt** 讀入兩個字串(每一字串皆不超過 50 字元)，判斷第一個字串是否為第二個字串的部分序列字串，若是則輸出 Yes 以及被刪除之部分序列字串，否則輸出 No。

範例輸入一：

bcfg gcbgcefgjvk

範例輸出一：

bcfg gcbgcefgjvk

Ans=Yes gcgejvk

範例輸入二：

bcfg gcagcefgjvk

範例輸出二：

bcfg gcagcefgjvk

Ans=No

-
20. 假設 N 個正整數數列是依據下面的方式來產生：

$X_1=1, X_2=2, X_3=3$

$X_i=(X_{i-3}+X_{i-2}+X_{i-1}) \% M + 1, \text{ for } i = 4 \text{ to } N$

假設 $M=12$ ，則產生的數列如下：

1 2 3 7 1 12 9 11 9 6 3 7 5 4 5 3 1 10 3 3

假設另外給定一個正整數 K，請撰寫程式找出包含 1, 2, 3, ..., K 的最小子數列長度以 $K=4$ 為例，最小的子數列為：

1 2 3 7 1 12 9 11 9 6 3 7 5 4

所以，長度為 14

請從 **e.txt** 中讀入 N, M, K 三個數字，接著輸出最小的子數列長度，若是找不到則輸出 No。

輸入範例 1：

20 12 4

輸出範例 1：

20 12 4

Ans=14

輸入範例 2：

20 12 8

輸出範例 2：

20 12 8

Ans=No

-
21. 猜數字：請撰寫一個程式從 **a.txt** 中讀入兩個數字字串（長度最長為 10 且數字不重複），並且判斷字串中的每個數字，若是兩個字串在同一個位置的數字都相同，則記載一個 A，若是兩個字串都有這個數字但是位置不同，則記載一個 B，當沒有 A 也沒有 B 時則輸出 **0A0B**，底下是輸入的範例：

14685

47653

輸出結果：

14685

47653

Ans=1A2B

-
22. 排序：請針對 **b.txt** 中的資料進行排序，並依題意由大至小或由小至大將排序結果印出，排序方式不限，排序後的資料間以一個空白間格分隔。

輸入檔說明：

輸入檔中包含兩行文字，第一行的數字為 0 或 1。0 代表輸出結果由小至大排列，1 則代表輸出結果由大至小排列。第二行則包含欲排序的數字，每個數字均為整數，數字個數不超過 100 個，每個數字間以空白間隔。

範例輸入：

0

3 2 5 7 8 1

範例輸出：

0

3 2 5 7 8 1

Ans=1 2 3 5 7 8

-
23. 數學大師歐勒找到了一個圓周率的無窮乘積計算式，如下所示：

$$\frac{\pi}{2} = \frac{3}{2} \times \frac{5}{6} \times \frac{7}{6} \times \frac{11}{10} \times \frac{13}{14} \times \frac{17}{18} \times \frac{19}{18} \times \frac{23}{22} \times \dots$$

在這個公式內所有的分子都是大於 2 的質數，分母則是不能被 4 整除且最靠近分子的偶數。請撰寫一個程式，可以從資料檔 **c.txt** 中讀入一個正整數 n 後，接著分別列出上面式子到第 n 項的分子與分母的乘積運算式，第一行輸出分子的乘積，第二行輸出分母的乘積運算式。

範例輸入一：

5

範例輸出一：

5

Ans=3*5*7*11*13

Ans=2*6*6*10*14

範例輸入二：

8

範例輸出二：

8

Ans=3*5*7*11*13*17*19*23

Ans=2*6*6*10*14*18*18*22

-
24. 請從資料檔 **d.txt** 讀入兩個數字 x 及 n 後計算下面運算式的結果：

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

輸出結果時只需要輸出整數部分，小數點後的結果無條件捨去。

範例輸入：

5

8

計算方法：

$$\sum_{k=1}^8 \frac{5^k}{k!} = \frac{5^1}{1!} + \frac{5^2}{2!} + \frac{5^3}{3!} + \frac{5^4}{4!} + \frac{5^5}{5!} + \frac{5^6}{6!} + \frac{5^7}{7!} + \frac{5^8}{8!} = 137.3072$$

範例輸出：

5

8

Ans=137

-
25. 雞兔同籠：有一個籠子裡裝了若干的雞和兔，數一數他們的頭，一共有 15 顆頭，數一數他們的腳，一共有 42 隻腳，請問雞和兔各有多少隻呢？

透過簡單的代數計算可以得知雞有 9 隻，兔子有 6 隻，他們的腳一共是：

$$9*2+6*4=18+24=42$$

請撰寫一個程式，可以從資料檔 **e.txt** 中讀入四個數字，分別表示第一種動物的腳數、第二種動物的腳數、頭的總數與腳的總數，然後輸出第一種動物的數目以及第二種動物的數目，若是解答不存在的話則輸出 No Solution。

範例輸入一：(第一種動物 2 隻腳，第二種動物 4 隻腳，頭一共 15 個，腳一共 42 隻)

2 4 15 42

範例輸出一：

2 4 15 42

Ans=9 6

範例輸入二：(第一種動物 2 隻腳，第二種動物 4 隻腳，頭一共 10 個，腳一共 42 隻)

2 4 10 42

範例輸出二：(注意：No Solution 的字串請與題目說明完全一致，以避免驗證失敗)

2 4 10 42

Ans=No Solution

-
26. 有一天小明設計了一個新的數值大小比較方法，主要依照數值的「位數」以及「數字加總取 10 的餘數」所轉換出的新數值來比較，舉例來說，有兩個數值 117 與 899，117 是 3 位數，數字加總是 $1+1+7=9$ ，轉換為新數值 39，899 也是 3 位數，數字加總是 $8+9+9=26$ ，取 10 的餘數是 6，因此形成新數字 36，由於轉換後的數值 $39>36$ ，所以原來的數值 $117>899$ 。

請從 **e.txt** 中先讀入一個數字 N ($N \leq 10$)，表示接著有多少個正整數，接著將每個正整數經過轉換並比較之後，輸出轉換後最大數值與對應的正整數（測試資料只會有唯一的最大值）。

範例輸入一：

5

215 69 923 327 188

上面的 5 個正整數轉換後的數值分別為

38 25 34 32 37

範例輸出一：

5

215 69 923 327 188

Ans=38 215

-
27. 小明想要安全的將一串由數字 0~9 所構成的字串傳給朋友，於是他想了一個方法，他把這個字串切成好幾段不固定長度的子字串，每一段都隨機使用一個位移編碼的方式來進行轉換，位移量為 1~9，並且每一段子字串的開頭加入一個字元 'C'，這樣朋友收到之後，就可以依照指定方式來還原出原始字串。底下是一個例子，假設原始字串如下：

042332300077139487945

小明將上面字串隨機切割為好幾個子字串如下：

04 23323000 771394 87945

每一段使用不同的隨機位移量進行編碼：

04 → (位移量 3) → 37

23323000 → (位移量 8) → 01101888

771394 → (位移量 2) → 993516

87945 → (位移量 7) → 54612

所以真正傳送給朋友的字串如下：

C337C801101888C2993516C754612

麻煩從資料檔 **d.txt** 中讀入這一段編碼的字串，然後解碼還原出原本小明想要傳送的字串。

範例輸入：

C337C801101888C2993516C754612

範例輸出：

C337C801101888C2993516C754612

Ans=042332300077139487945

-
28. 保齡球運動的計分方式很特別，每一局有 10 格，一連串的擊瓶結果用 0~9, X 與 / 來標示。0~9 表示擊倒瓶數，如果標記 X，表示一次擊倒，該格的分數=10+接續的兩個數字，如果標記 /，該格的分數=10+下一個數字。不是 X 或者 / 的情況，都是直接加上數字來累加分數。底下是一個例子：

每球得分 6 2 8 / 5 1 X 7 2 X X 7 2 6 / 8 1

實際得分 6 2 8 7 5 1 19 7 2 27 19 7 2 6 12 8 1

第一個 /：前一球 8 瓶，所以該球 2 分，加上後一球 5 分，得分 2+5=7

第一個 X：全倒 10 分，加上後面兩球，得分 10+7+2=19

第二個 X：全倒 10 分，加上後面兩球，第一球 10 分，第二球 7 分，得分 10+10+7=27

第三個 X: 全倒 10 分, 加上後面兩球, 第一球 7 分, 第二球 2 分, 得分 10+7+2=19

第二個/: 前一瓶 6 分, 所以該球 4 分, 加上後一球 8 分, 得分 4+8=12

將上面實際得分累加就可以得到目前分數:

得分累加: 6 8 16 23 28 29 48 55 57 84 103 110 112 118 130 138 139

請撰寫一個程式可以從 **e.txt** 先讀入一個數字表示有幾組測試資料, 接著依序輸出分數累加結果, 若是還無法累加得知結果則輸出 'Unknown'。

範例輸入:

```
3
6 2 8 / 5
6 2 8 / 5 1 x 7
6 2 8 / 5 1 x 7 2 x x 7 2 6 / 8 1
```

範例輸出:

```
3
6 2 8 / 5
6 2 8 / 5 1 x 7
6 2 8 / 5 1 x 7 2 x x 7 2 6 / 8 1
Ans=28 Unknown 139
```

-
29. 小明有一天拿到一份資料, 是由 0 跟 1 組成的二維陣列, 表示一個地圖上面有道路的地方, 0 表示沒有馬路, 1 表示有馬路, 接著他想要找出最長的一條路, 這條路只能是水平線, 垂直線或者對角線, 請撰寫一個程式來協助小明解決問題。

請從 **e.txt** 中先讀入寬與高 (皆不會超過 30), 接著讀入這個二維陣列, 然後輸出最長的一條路。

範例輸入:

```
6 14
01001110001000
00100100010000
01110110111000
00001000000100
00000100001010
00111110010000
```

範例輸出:

```
6 14
01001110001000
00100100010000
01110110111000
00001000000100
00000100001010
00111110010000
```

Ans=6

-
30. 小明有一天外出在地上撿到一張信用卡, 為了確認該卡片號碼是否造假, 小明依據下面的演算法 (Luhn Formula) 進行計算:

1. 假設卡號如下:

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 2

2. 反轉所有數字, 結果如下:

2 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1

3. 將偶數位置的字元乘以 2，結果如下：

2 10 4 6 2 2 0 18 8 14 6 10 4 6 2 2

4. 將每個單一字元的數值進行加總，結果如下：

$2+1+0+4+6+2+2+0+1+8+8+1+4+6+1+0+4+6+2+2=60$

5. 加總結果如果是 10 的倍數，表示卡號是有效的 (valid)，否則就是無效卡號 (invalid)。

請撰寫程式可以從 **e.txt** 資料檔，先讀入測試資料筆數 N，接著，針對下面 N 行卡號，逐一讀入，並輸出對應的 valid 或 invalid。

範例輸入：

2

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 2

5 5 2 0 0 4 0 0 4 7 9 4 3 4 0 9

範例輸出：

2

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 2

5 5 2 0 0 4 0 0 4 7 9 4 3 4 0 9

Ans=valid invalid

31. 請撰寫一個程式，從資料檔 **e.txt** 讀入一個英文句子 (由 a-z 的大小寫與空白組成，不含句點)，裡面有好幾個單字，接著以最長單字為最大寬度，將這些單字輸出在一個由星號字元的長方形中。

範例輸入：

CYUT is the top 1000 university

範例輸出：

CYUT is the top 1000 university

*CYUT *

*is *

*the *

*top *

*1000 *

university

32. 字串加密：試寫一個程式可以讀入一個字串 (長度最大為 100)，接著將字串中的英文字元 ('a'-'z' 或 'A'-'Z')，依照其順序數值進行加總 ('a' 與 'A' 的順序為 1, 'b' 與 'B' 的順序為 2，依此類推)，加總結果取 26 的餘數視為密鑰，也是每個英文字元的位移量，假設位移量是 2，則 'a'-'z' 的加密結果如下：

原始: abcdefghijklmnopqrstuvwxyz

加密: cdefghijklmnopqrstuvwxyzab

請從資料檔 **e.txt** 中讀入一行字串，將字串中的大小寫英文字元都進行加密，其餘字元則是原封不動，輸出加密後的字串。

範例輸入：

It is never too late to learn, just do it.

範例輸出：(所有英文字元依據其順序的加總值為 412，取 26 的餘數為 22)

It is never too late to learn, just do it.

Ans=Ep eo jaran pkk hwpk hawnj, fqop zk ep.

-
33. 撲克牌配對是常見的遊戲，假設有 16 張撲克牌，內含 8 個對子(數字相同)，接著玩家一次翻開二張，若是不一樣，就翻回去，若是一樣，就不用翻回去，直到將所有牌翻開為止。

請從檔案 **e.txt** 中讀入第一行的數字 N ($1 \sim 20$)，表示有多少個對子，接著第二行讀入 $2*N$ 個數字。假設玩家具有過目不忘的能力，翻開過的都會記住，請依序從頭到尾來翻牌，然後依據上面翻牌的規則，計算需要翻幾次才能完成遊戲(翻開與翻回去都各算一次)。底下是一個實例：

範例輸入：

3

4 8 4 10 8 10

翻牌配對的過程如下：

先翻開前兩張 4 8 -> 2 次，配對失敗翻回去 -> 2 次

接著翻開第 2 個 4，也記住已經翻過的第 1 個 4，配對成功 -> 2 次

接著依序翻開 10 8 -> 2 次，配對失敗翻回去 -> 2 次

接著翻開第 2 個 8 與第一個 8，配對成功-> 2 次

接著翻開第 2 個 10，也記住已經翻過的第 1 個 10，配對成功 -> 2 次

合計翻開與翻回去共 14 次

範例輸出：

3

4 8 4 10 8 10

Ans=14