

重要的競賽指南!!



請仔細的閱讀以下的指南。它包含了將如何打包和提交你的答案給裁判的重要資訊。如果對於這份指南有任何問題，請在競賽開始前提問。

程式輸入/輸出:

需要互動的程式(task)應該要即時在螢幕上(STDOUT)要求使用者輸入，並且從鍵盤/螢幕讀取輸入(STDIN)。

對於不需要互動的程式(task)，你會有二種選擇來讀取輸入資料。在一些題目當中，名為**probXX.txt**的檔案將會被提供來作為樣本輸入，'XX'代表問題的號碼。

你的解答可能會需要有系統地從檔案 **probXX.txt** 讀取輸入。也就是說使用程式語言中的File I/O架構。

大部分的問題會接受直接由鍵盤來輸入(STDIN) 來取代檔案運作。對於需要很多輸入的程式，這樣就變得很冗長乏味。然而，一個簡單的方式是在執行的時候借由導入檔案的內容資料到你的程式中。例如：一個名為 **prob01.txt** 的檔案可以用下面的語法直接從STDIN輸入到你的程式當中：

```
%> java prob01 < prob01.txt
%> java -jar js.jar prob01.js < prob01.txt
%> python prob01.py < prob01.txt
%> prob01.exe < prob01.txt
```

在這個例題裡，你正執行著 **prob01** 以及把 **prob01.txt**檔案中的資料傳送到你程式中的STDIN。你的程式行為就會如同你從鍵盤鍵入你的輸入一樣。

提示：當使用鍵盤直接輸入時，請使用 'Ctrl-Z' <return> 來結束你的程式輸入。

所有程式的輸出應該都要傳送到螢幕上(STDOUT)。

提交你的程式

Interpreted Programs (JAVA, JavaScript, Python)。你的程式一定要用 **probXX.java** / **probXX.js** / **probXX.py**來命名，'XX'代表問題的號碼。請僅提交原始碼(.java、.js 或 .py)。Java的主要類別一定要命名為 **probXX**。請注意大小寫。所有主要和支援類別都應該包含在預設的(或匿名的)套件裡。

Native Programs (C, C++... etc)。你的程式應該用 **probXX.exe**來命名，'XX'代表問題的號碼。

強烈地建議你在開始競賽之前，

先提交問題#0 (在下一頁上) 以確保你的編譯環境和裁判的是相容的。

前言

這個題目的主要目的是讓每一個參賽團隊試著遞交一個測試程式，以確保產生出來的結果可以在評審的電腦上正確執行。強烈建議每一個參賽團隊先遞交這題。

這題改編自經典的「Hello World!」程式，請印出「Hello HP CodeWars 2016 Taipei!」。

輸入

[這題沒有輸入。]

輸出

Hello HP CodeWars 2016 Taipei!

前言

銀河帝國覆亡了 30 年後，其餘燼中誕生了第一軍團，邪惡的第一軍團無惡不作，身為反抗勢力的你接獲指令要尋找天行者路克抵抗第一軍團，現在尋找天行者路克的最後一塊拼圖就在 R2D2 身上，但 R2D2 經過長期休眠後電腦編碼系統發生了錯誤，請聰明的你/妳為 R2D2 寫出校定程序吧！

下面為 R2D2 電腦編碼系統中錯誤的列表:

原始編碼	錯誤編碼	原始編碼	錯誤編碼
A	Z	a	z
B	Y	b	y
C	X	c	x
D	W	d	w
E	V	e	v
F	U	f	u
G	T	g	t
H	S	h	s
I	R	i	r
J	Q	j	q
K	P	k	p
L	O	l	o
M	N	m	n
N	M	n	m
O	L	o	l
P	K	p	k
Q	J	q	j
R	I	r	i
S	H	s	h
T	G	t	g
U	F	u	f
V	E	v	e
W	D	w	d
X	C	x	c
Y	B	y	b
Z	A	z	a

當輸入為以下範例中的輸入字串時，再經過你的校定程序校正後，輸出結果為輸出字串。

當輸入字串內的字元如不包含在錯誤列表時，則視為正確的編碼，無須校正。

輸入

輸入為 80 個字元以內的 ASCII code 字串。

實例1：“Nzb gsv ulixv yv drgs blf.”

實例2：“l2W2 Dv'iv wllnvwl”

實例3：“R szez z yzw uvvormt zylfg gsrh.”

輸出

程式必須輸出輸入字串校正後的結果。

實例1：“May the force be with you.”

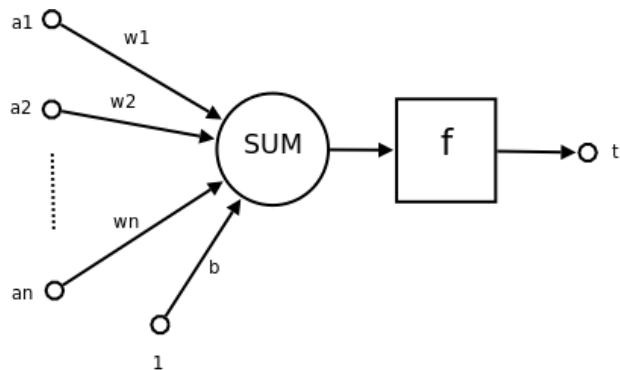
實例2：“R2D2 We're doomed!”

實例3：“I have a bad feeling about this.”

前言

設計一個神經元 (Neuron)

利用類神經網路作為基礎的深度學習演算法在近幾年有很多成功的案例，尤其是應用在電腦視覺以及語音辨識的應用上，都已經達到了商用化的標準，而一組類神經網路，其實就是由大量的神經元所組成。



上圖為最基本的神經元示意圖， $a_1 \sim a_n$ 為輸入向量的各個分量， $w_1 \sim w_n$ 為神經元各個突觸的權值， b 可視為一調整的常數， f 為傳導函數，通常為非線性函數，我們可以假設採用 $\text{hardlim}(x)$ ， $\text{hardlim}(x)$ 的定義為

$$\text{hardlim}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

所以，一個神經元的數學表示可以寫成

$$t = f(a_1 * w_1 + a_2 * w_2 + \dots + a_n * w_n + b)$$

輸入

第一行輸入 n 跟 b 的數值，中間以空白分隔。

第二行輸入 a_1 到 a_n 的數值，每個數值中間以空白分隔。

第三行輸入 w_1 到 w_n 的數值，每個數值中間以空白分隔。

實例1:

2 0.12
10 3
0.1 -0.4

實例2:

3 15
143 -29 -17
0.023 0.18 0.76

輸出

請輸出計算後的 t 值。

實例1:

0

實例2:

1

前言

大數據分析是國內外非常熱門的領域，資料科學家們藉由資料收集、整理、儲存、建模、分析到展示，從大量的真實資料中找出所有可能的關聯性，藉此可以延伸運用到預測、決策分析、人工智慧、自動化等各行各業及各種領域，探究人們過往所不瞭解的事情。

對於大數據資料分析平台中，其中有一個關鍵的部份，就是必須能夠處理結構、非結構及半結構化的資料，簡單解釋一下結構、半結構、非結構化資料，結構化的資料是指可以預先知道欄位長度的資料，如一般會放進關連資料庫處理的內容；半結構化資料則比較像 `key, value` 的索引，每一筆資料的各個欄位長度、一筆資料總長度和筆數都不是固定的；而非結構化資料則比較屬於「內容」資料，如：圖片、網頁等，當然，結構資料是同質性的，越非結構資料的資料間相關性就越低，如：學生資料（結構資料）、ATM 交易紀錄（半結構化資料）和整個網站資料（圖片、交易紀錄、程式），為了使資料能結構化以便更容易處理，則需要把一些非結構化的資料轉為半結構化或是結構化的資料，或是把半結構化資料轉為結構化資料。

所以，我們需要一個程式能夠將非結構化資料在「整理」階段，做一些預先的處理。如：

[a.txt]

name=David,age=20,city=Taipei

NTU address=No 1, RoseFu Rd 4th Sec, Taipei, Taiwan, 10617

可以知道 Taipei 出現了 2 次

輸入

輸入一個文字檔案，找出指定文字出現的資料。大小寫字母視為不同字母。

實例1：

a.txt "Taipei"

實例2：

a.txt "David"

實例3：

a.txt "Foobar"

輸出

實例1： 2

實例2： 1

實例3： 0

前言

雞兔同籠問題是七年級常計算的問題，兔子有四隻腳、雞有兩隻腳，假如給定雞和兔的數量，以及雞兔共有的腳數，則我們可以計算出，雞有幾隻、兔有幾隻。

例如：雞和兔共有 20 隻，腳共有 70 隻，則可以計算出兔子有 15 隻，而雞有 5 隻。

輸入

雞和兔的數量，以及腳總共的數字

實例1：

20 70

實例2：

15 30

實例3：

60 60

輸出

如果有解，則輸出兔子和雞的數量。若無解則輸出"None"。

實例1： 15 5

實例2： 0 15

實例3： None

前言

柵欄(Rail Fence)演算法

柵欄加密 (Rail Fence) 演算法是利用改變字元排列的次序，將字串重排加密的方法。一個比較簡單的作法，可以將明文字串依橫列的順序，由左向右書寫，為了避免混淆，我們把空白替換成 *，並且同樣地將句尾加上 * 以填滿所有空格。然後，依照直行順序由上而下依序取出字串，如下圖所示，將 "This is an easy question" 加密後，取出來的字串就會變成 "Tneh*sietsai*soiyns***q*au*"。

1	2	3	4	5	6	7	8	9
T	h	i	S	*	i	s	*	a
n	*	e	A	s	y	*	q	u
e	s	t	l	o	n	*	*	*

輸入

輸入一個字串，前後以雙引號標示，另外再輸入直列(Column)的數目。

實例1: "The tree is tall" 5

實例2: "May the Code be with You" 6

實例3: "HPE manufactures the best server products in the world" 10

輸出

請輸出加密後的字串，前後以雙引號標示。

實例1: "Trslhe**eet***a*til**"

實例2: "Me*ta*bhyCe**o*Ytdwoheiu"

實例3: "Hc*esoPtbr*rEue*il*rspndmetr**as*ot*n*sdh*uteue*fhrc**aevtw**"

前言

美國太空總署去年發現一個命名為克卜勒 452b 的行星，地處天鵝座，與地球相距一千四百年光年，體積比地球大百分之 60。假設克卜勒公轉一年要 385 天，並且每 5 年一閏 (第 5 年為閏年)，每 100 年不閏，每 500 年又閏，一年有 12 個月，假設登陸第 1 天的日期設為 CodeWars 1 月 1 日星期一，一週仍有 7 天 (星期一到星期六和星期日, 即數字 1-7)，3 月為 33 天，其他月為 32 天，閏年時 9 月為 33 天。

輸入

輸入任一登陸的年月日

實例1：

5 10 1

實例2：

500 1 1

實例3：

500 10 1

輸出

輸出該日是一週的第幾天

實例1： 4

實例2： 5

實例3： 1

前言

在數學理論中，將一個正整數表示成其質因數的乘積，稱為質因數分解；舉例而言，我們可以將 12 分解成 $2^2 \times 3$ 。有趣的是，雖然質因數分解的概念很容易理解，但當欲進行分解的數值過大時，便會衍生出計算繁複且耗時的問題，因而此方法被廣泛地運用於密碼學領域，其中 RSA 演算法就是一個最具代表性的例子。現在就讓我們踏出探索密碼學的第一步 – 給定一個整數，請輸出此數的質因數分解。

輸入

一個需做質因數分解的整數 n ， $n < 2^{31} - 1$ 。

實例 1: 12

實例 2: 168

實例 3: 1

輸出

質因數分解的結果，輸出的行數視分解的結果而定，並按照質因數的數值由小至大依序印出，其中每一行都包含了分解後的質因數及其冪數。若給定的整數無法進行質因數分解，請直接輸出 0。

實例 1:

2 2

3 1

實例 2:

2 3

3 1

7 1

實例 3:

0

問題 8

巧克力分送

15 分

前言

兒童節到了，HP 決定邀請 N 個小朋友過來玩遊戲，遊戲結束後，HP 會根據小朋友的分數發送巧克力當獎勵。現在小朋友不依照分數的高低排成一直線，每個小朋友至少要得到一個巧克力，分數高的小朋友得到的巧克力數量必須大於他左右兩邊的小朋友，分數相同的小朋友不一定會拿到相同數量的巧克力，請問 HP 發送的最小巧克力總數是？

下面為驗算範例：

	小朋友 A	小朋友 B	小朋友 C	小朋友 D	小朋友 E
分數	10	20	30	30	30
所得糖果	1	2	3	1	1
糖果最小總數	$1+2+3+1+1 = 8$				

輸入

程式必須輸入一組小朋友的分數序列

實例 1: 40 30 30 10 20 10

實例 2: 10 20 30 20 30

輸出

程式必須輸出最小的巧克力總數

實例 1: 9

實例 2: 9

前言

細想一個兩位數介於 10 到 99 之間的 8x8 矩陣，如下所

68	41	59	43	42	43	32	85
52	87	97	72	77	50	97	19
72	23	77	87	62	32	92	46
98	25	76	78	33	18	70	30
29	22	59	46	81	36	18	25
19	26	33	76	95	61	19	79
46	44	87	45	11	98	20	62
99	91	14	70	30	75	34	98

你設計電腦程式將會從單行輸入矩陣的所有數字，按照每列排序。一旦所有 64 個號碼都被輸入完，它會掃描整個矩陣並且發現連續兩個數字是遞增排序或遞減排序，位於水平的或垂直的。

68	41	59	43	42	43	32	85
52	87	97	72	77	50	97	19
72	23	77	87	62	32	92	46
98	25	76	78	33	18	70	30
29	22	59	46	81	36	18	25
19	26	33	76	95	61	19	79
46	44	87	45	11	98	20	62
99	91	14	70	30	75	34	98

輸出將包含兩個數字。先用水平發現成對數字然後用垂直發現成對數字。在有三個連續號碼的情況下，將代表兩個不同例子。

輸入

在一行中輸入所有 64 個數字矩陣，按列水平排序並用空格隔開數字。

例 1: 68 41 59 43 42 43 32 85 52 87 97 72 77 50 97 19 72 23 77 87 62 32 92 46 98 25 76 78 33 18 70
30 29 22 59 46 81 36 18 25 19 26 33 76 95 61 19 79 46 44 87 45 11 98 20 62 99 91 14 70 30 75
34 98

例 2: 49 84 79 56 28 43 67 36 37 40 68 72 55 62 22 82 60 56 57 46 39 57 73 65 76 74 70 51 40 75 56
45 52 63 32 80 64 53 74 34 48 55 51 54 45 44 35 51 61 45 33 61 77 60 85 68 34 67 42 69 52 68
52 47

輸出

返回搜索結果

例 1: 2 3

例 2: 2 1

前言

在反抗軍戰士間最近流行一款絕地武士卡牌手機遊戲。遊戲中的絕地武士卡片分成四個階級，由高至低為：Legendary (L), Jedi Master (M), Jedi Knight (K), 以及 Apprentice (A)。愈高級的卡片愈稀有，抽到的機率也愈低。

在已知四種卡片的出現機率下，請幫我們辛苦的反抗軍戰士們計算出能夠抽到各種卡片需要的次數。

抽不到卡片的機率是這樣計算的: $(1 - \text{卡片出現機率})$ 與 (抽取次數) 的乘冪。隨著次數的增加，抽不到卡片的機率會愈來愈低，當它低到 $< 1\%$ 的時候我們就認定抽這麼多次一定可以抽到。

舉例來說，假如 A 卡的出現機率是 80%，抽取兩次時抽不到 A 卡的機率是 $(1-0.8)$ 的 2 次方 = 4%，抽取三次時抽不到的機率是 $(1-0.8)$ 的 3 次方 = 0.8%。所以我們認為只要抽三次就一定可以抽到一張 A 卡。

題目會給出四種卡片的機率，請回答四種卡片的抽取次數。

輸入

輸入是四種卡片依 L M K A 的出現機率，單位是百分之一，總和必然為 100。

實例1: 1 5 34 60

實例2: 2 10 30 58

實例3: 1 3 25 71

輸出

請依照 L M K A 的順序印出需要抽取的次數

實例1: 459 90 12 6

實例2: 228 44 13 6

實例3: 459 152 17 4

前言

賓果遊戲是一款廣受歡迎且風靡全球的遊戲。在這個遊戲中，我們有一個印著正整數的 $n \times n$ 棋盤，以及所有主持人抽出的幸運號碼，而你的任務就是在棋盤上標記這些號碼，並找出這些號碼能在棋盤中連成幾條線，包括了水平、垂直以及斜向成線三種。你需周詳地考慮各種不同的狀況：棋盤上可能會出現重複的數字、所抽出之幸運號碼或許不在棋盤上等等。除此之外，你需將棋盤上的特殊星形符號 "*" 視為一個已標記的棋盤格子，藉此增加連線的機會。

輸入

輸入包含數行：第一行為棋盤的邊長 n ，接下來的 n 行將一一列出棋盤上的數字，範圍介於 1-999，

最後一行則是由主持人抽出的一連串幸運號碼，而這些號碼將以空格分開表示並以 0 作為結尾。

實例 1：

```
5
8 3 23 19 14
6 21 1 9 10
11 18 4 24 15
17 2 7 22 13
5 12 16 20 25
8 22 4 21 11 6 25 5 17 0
```

8	3	23	19	14
6	21	1	9	10
11	18	4	24	15
17	2	7	22	13
5	12	16	20	25

實例 2：

```
3
5 22 8
27 * 3
16 57 12
3 8 16 57 12 27 5 22 0
```

5	22	8
27	*	3
16	57	12

實例 3：

```
3
* 6 *
21 * 49
* 18 *
21 0
```

*	6	*
21	*	49
*	18	*

輸出

請輸出在棋盤上所能連成直線的數量。如果沒有可以連線的情形，請輸出 0。

實例 1： 2

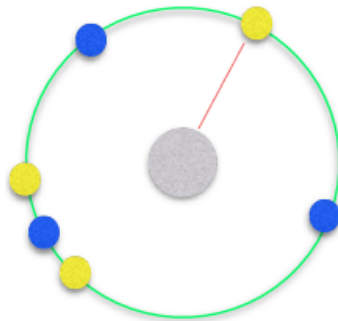
實例 2： 8

實例 3： 3

前言

第一軍團在銀河的深處建造了弑星者基地用來摧毀不肯合作的共和國行星或是反抗勢力。弑星者基地在摧毀一顆行星之前要先吸乾另外一顆發光恆星的能量並將其轉換成光束射向要摧毀的行星，但是沒辦法累積超過一顆恆星的能量。赫斯將軍下令要摧毀三顆共和國行星，並即刻開始吸光恆星的能量。反抗勢力在緊要關頭接到了密報，上面記錄了弑星者基地的攻擊目標以及作為能量來源的恆星位置。

弑星者基地每發射一發光束之前要先吸乾一顆恆星而且一顆恆星的能量剛剛好足夠摧毀掉一顆共和國行星。很湊巧的三顆共和國行星跟三顆發光恆星都在同一個平面上，六顆星星剛好可以形成一個正圓。弑星者基地位於此正圓的圓心。由於弑星者基地的發射口必須要對準星星才可以吸收能量或是發射光束，所以弑星者基地必須要自行旋轉來對準星星。弑星者基地每轉動一圈需花費 720 秒，但是可以自由地順時針或是逆時針轉動。弑星者吸收一顆恆星的能量需要花費 60 秒，摧毀一顆行星需花費 5 秒。反抗勢力的莉亞將軍需要知道第一軍團最少要花多少時間來摧毀三顆共和國行星，請根據密報提供的資訊來計算出來吧！



輸入

星球在圓周上的位置可以用角度來表示(0~360)，弑星者基地的發射口起始位置對準 0 度。輸入為 6 個數字(範圍在 0~360)，前三個代表發光恆星的位置，後三個代表共和國行星的位置。

1. 0 90 180 10 100 190
2. 20 40 270 280 90 120
3. 0 10 20 330 340 350

輸出

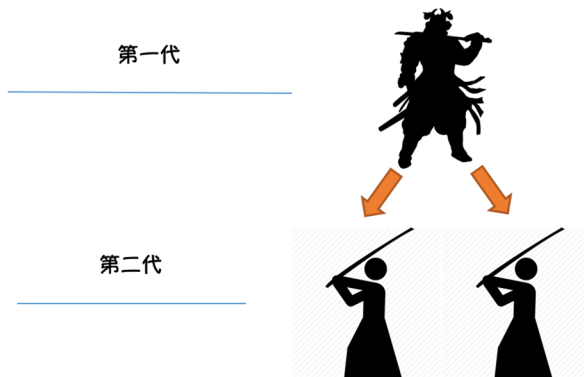
摧毀三顆共和國行星所需的最少秒數

1. 575
2. 955
3. 495

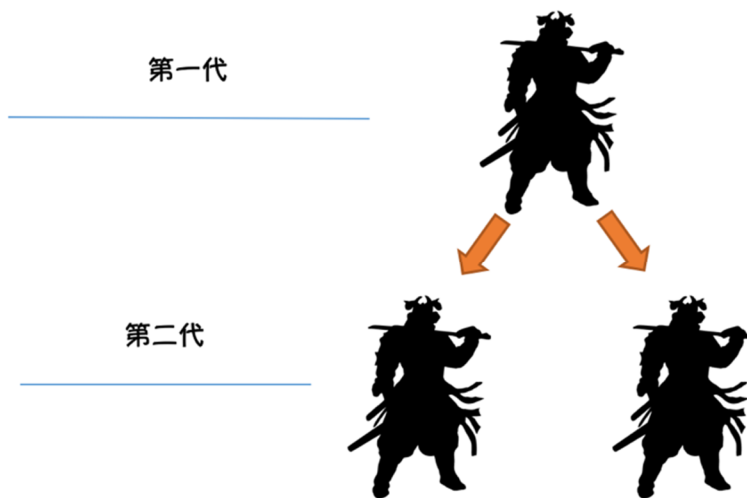
前言

絕地武士家族, 為了維護自身流派系的素質, 絕地武士宗師規定了幾項收受弟子的規定, 如下列所述

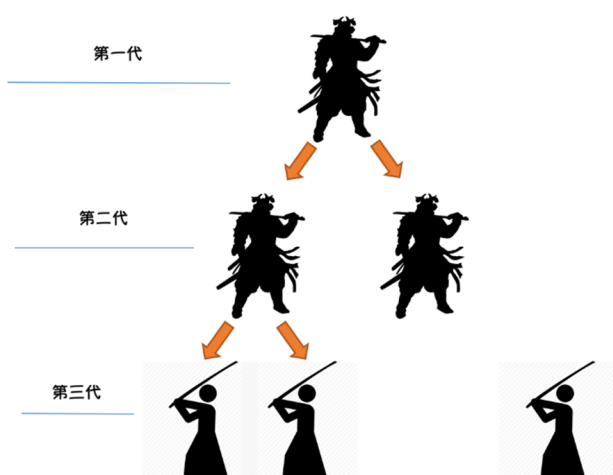
1. 一個絕地武士大師至多只能收受兩名徒弟



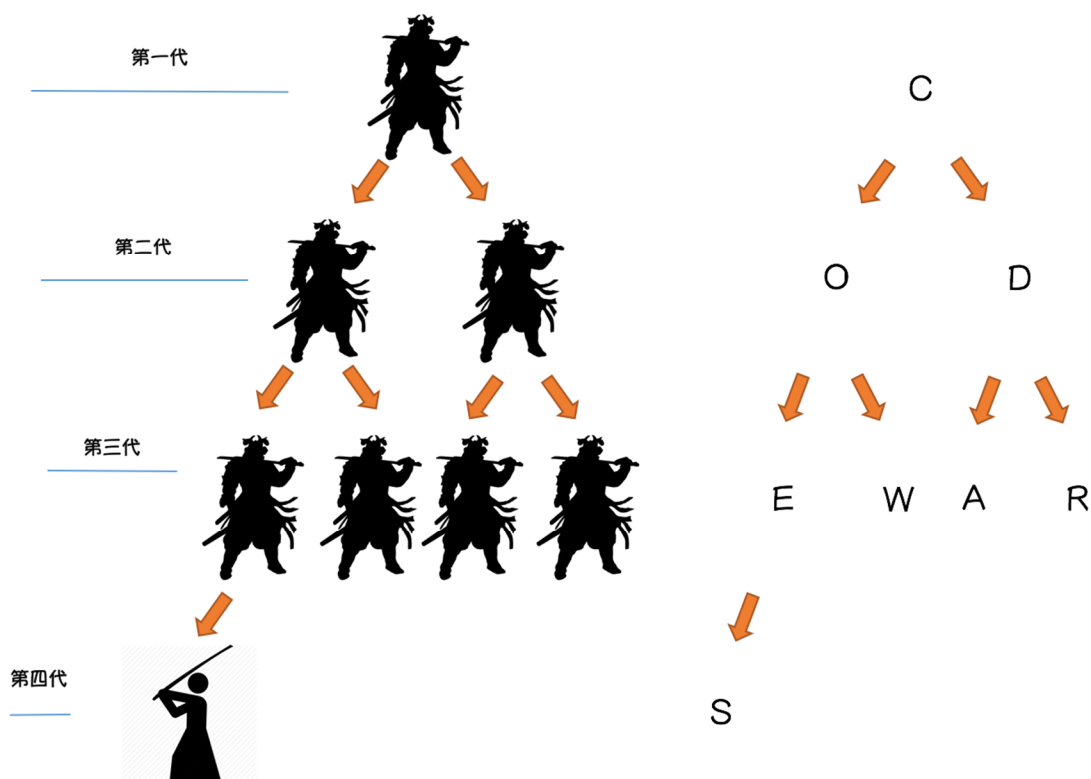
2. 當所有同一代絕地武士大師將其所有弟子收滿, 其徒弟一代即可轉換身份而成為絕地武士大師, 並可以開始收受徒弟



- 絕地武士收受徒弟制度嚴明, 必須由輩份順序, 依序收受弟子, 也就是越先早加入其流派的弟子輩分越高, 同一代的絕地武士大師, 輩分最高的先收滿兩名徒弟後, 次弟子始能開始收受徒弟, 以此類推



- 同一派別的絕地武士中, 會有一串記述著自己派別各個輩分代號的心法, 假設絕地武士招收的順序是“C, O, D, E, W, A, R, S” 如左下圖樹狀圖可以表示為右下圖樹狀圖, 則右下樹狀圖會表示成“C, O, E, S, W, D, A, R”以成為該派別的代稱, 絕地武士大師和絕地武士或許會有同樣代號



輸入

輸入為派別之心法(並非招收的順序),派別以逗號加空白分開,輸入只有大寫

1. C, O, E, S, W, D, A, R
2. H, E, L, P, A, E, C, K, W, T, A, R, T, D
3. S

輸出

輸出為目前沒有任何弟子的絕地武士大師或是絕地武士之代號, 並依輩分由高至低輸出, 逗號加空白分開, 輸入只有大寫

1. W, A, R, S
2. P, A, C, K, A, R, D
3. S

前言

第一軍團在一個古老的藍色星球上發現了一種從古代流傳下來的小遊戲。玩法是在固定大小的盤面上消除相同種類寶石，以下圖為例：

G	G	Y	G	Y	G	R	R
B	B	Y	B	Y	G	R	R
B	G	Y	R	R	B	G	B
B	B	R	R	B	R	G	B
R	G	R	B	B	Y	B	B

為了方便說明，我們把最左下角的格子定為(0,0)，右下是(7,0)，左上是(0,4)，而 R, G, B, Y 代表紅，綠，藍，黃四種顏色的寶石。玩法是這樣的：

1. 選擇一個待消除的格子，如果這個格子的上下左右四格當中有相同顏色的寶石，則會一起被消除。每個被消除的格子上下左右如果有同樣顏色的寶石也會一起被消除，直到無法消除為止。
2. 空格上方的寶石會落下，如果有空的空行，空行右側的寶石全部往左移動一行。
3. 回到第一步，進行到所有珠子都消除完畢或是無法消除為止。

以上圖來說，如果我們選擇消除(1,1) 會變成這樣

G	G	Y	G	Y	G	R	R
		Y	B	Y	G	R	R
	G	Y	R	R	B	G	B
		R	R	B	R	G	B
R	G	R	B	B	Y	B	B

		Y	G	Y	G	R	R
		Y	B	Y	G	R	R
	G	Y	R	R	B	G	B
G	G	R	R	B	R	G	B
R	G	R	B	B	Y	B	B

這時候再消除一次(1,2) 會變成這樣：

		Y	G	Y	G	R	R
		Y	B	Y	G	R	R
		Y	R	R	B	G	B
		R	R	B	R	G	B
R		R	B	B	Y	B	B

然後往左移一排

	Y	G	Y	G	R	R	
	Y	B	Y	G	R	R	
	Y	R	R	B	G	B	
	R	R	B	R	G	B	
R	R	B	B	Y	B	B	

請依照這樣的規則，印出指定步數後剩下的各色寶石數目。所指定的走數有可能會在無法消除的寶石上。

輸入

輸入是一組 8X5 的盤面，以及一連串待消除的格子座標，輸入 8 5 時結束輸入，依輸入順序由上而下排列：

實例1：

```
GGYGYGRR
BBYBYGRR
BGYRRBGB
BBRRBRGB
RGRBBYBB
1 1
1 2
8 5
```

輸出

請依 R G B Y 順序印出剩下的寶石數量

實例1： 11 5 9 6

前言

在工商業蓬勃發展的今日，自動販賣機在城市中幾乎隨處可見。身為一位智慧型自動販賣機的貼心設計師，在找零時你需提供最少的零錢個數以減輕顧客攜帶的負擔。在知道須找零的總數以及自動販賣機中各種面額的硬幣後（在此假設各種零錢的個數皆非常充足），請輸出機器最少需提供幾枚硬幣以完成找零。若現存的零錢種類無法完成找零，請在螢幕上輸出-1。

輸入

輸入共兩行：

第一行為所需找零的總數；

第二行則列出在販賣機中各種硬幣的面額，這些面額將以空格分開表示並以 0 作為結尾。

實例 1：

```
75
1 5 10 20 50 0
```

實例 2：

```
30
15 1 25 0
```

實例 3：

```
59
3 9 15 0
```

輸出

請輸出最少需提供幾枚硬幣以完成找零。

實例 1：

```
3
```

實例 2：

```
2
```

實例 3：

```
-1
```

前言

歡迎使用 HPE Simple file system，就如同你所用的檔案總管一樣
使用者可以刪除、新增或讀取檔案。
但是很不巧，系統並沒有處理得很好，導致大家檔案一團亂。
因此我們決定重寫一個系統來管理大家的檔案。
當然我們也要告訴使用者操作成功或失敗，如果操作成功就給輸出
success，如果失敗就輸出 fail

因為是 Beta 版關係，所有指令只會對檔案做操作，以下是指令：

Read：讀取檔案
Write：修改檔案
Add：新增檔案
Del：刪除檔案

檔案與資料夾有以下屬性

- 名稱：檔案的名稱
- 資料型態：d 表示資料夾，f 表示檔案
- 其他人使用權限：擁有者可以透過屬性控制別人是否有修正的權利。
 - 檔案部分
 - rw 表示其他人可以讀取也可以修改
 - r- 表示其他人可以讀取不可修改
 - 表示其他人不可讀取也不可修改
 - 資料夾部分
 - rw 表示其他人可以讀取/新增/刪除
 - 如果沒有 r 的權限，其他人無法做任何的操作 (讀取/修改/新增/刪除/)
- 擁有者：建立這個檔案/資料夾的人，擁有者可以有任意權限
- 資料夾：代表檔案所在的目錄

所以會如以下顯示一個檔案

TEST,f,r-,a1,DESKTOP

這是一個名為 TEST 的檔案，由 a1 所建立，其他人只能讀取這個檔案
(TEST 在 DESKTOP 這個資料夾下)

以下為實際範例：

a1,read,TEST,DESKTOP => success

(因為檔案是 a1 建立的)

b1,write,TEST,DESKTOP=> fail

(因為檔案是 a1 建立的，而且不允許其他人修改)

為了簡化問題，請注意以下事項

1. 所有的資料夾跟檔案名稱 “ 絕對不會重複”
2. 不用考慮複雜的檔名問題, 檔名跟資料夾可包含大寫 A-Z 和”.”
3. 新增檔案時，權限一律為 rw。
4. 無須考慮最初始的檔案系統狀態是否合理

輸入

輸入有兩個部分，第一個部分是原始資料的結構，[Folder_start][Folder_end]區塊為原本的資料夾，[File_start][File_end]區塊為檔案。

第二的部分為使用者使用情形，區塊為[Operation_start][Operation_end]，可能包含錯誤的操作。所有資料間都以逗號(,)做為區隔。

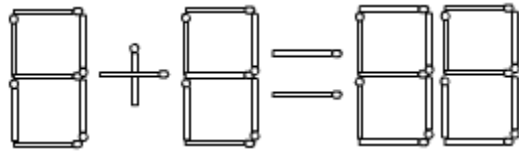
因為還是測試版本，所以使用者只有兩個工程師：a1, b1

```
[Folder_start]
DESKTOP,d,rw,b1
AFOLDER,d,rw,a1
[Folder_end]
[File_start]
A.TXT,f,rw,a1,DESKTOP
[File_end]
[Operation_start]
a1,write,A.TXT,DESKTOP
b1,write,A.TXT,AFOLDER
a1,read,A.TXT,DESKTOP
a1,del,QQQ,DESKTOP
[Operation_end]
```

輸出

```
Success
Fail
Success
Fail
```

前言



移動火柴棒（最多 3 2 根火柴棒）是一個休閒的遊戲，藉由移動火柴棒，找出合理的運算式，例如： $5 - 3 = 10$ ，可以藉由移動 2 根火柴棒成為 $9 - 9 = 0$ 或是 $5 + 3 = 8$ 。

請注意：

- "0" 用 6 根，"1" 用 2 根，"2" 用 5 根，"3" 用 5 根，"4" 用 4 根，"5" 用 5 根，"6" 用 6 根，"7" 用 3 根，"8" 用 7 根，"9" 用 6 根，"+" 用 2 根，"-" 用一根，移動火柴棒不包含增加或減少火柴棒。
- 輸入和輸出不能超過 6 個字。
- 第一位數和第二位數只能有一個數字，結果可以有兩個數字。
- 如果運算式不足六個字元，請在結果的數字前面補上空白。
- 火柴棒只能移到空的位置。
- 火柴棒不能移到曾經有火柴棒的地方。

合理運算式包含： $3 + 3 = 6$ （加法）， $9 - 9 = 0$ （減法）， $9 - 9 = 0$ （空白可在尾端），不合理運算式包含： $9 * 9 = 81$ （只有加減）， $9 + 0 = 09$ （答案不能有開頭 0）。

輸入

輸入一個 6 個字元的運算式，不足 6 個字元的運算式會在等號後補上空白，例如 "3+0= 6"。等號後面補一個空白。另外頭尾加上 "（因 "=" 和 " " 會被系統攔截，所以會加 quote）。

實例 1：

"3+0= 6"

實例 2：

"0+0= 0"

實例 3：

"9+9=11"

輸出

輸出移動 2 根火柴棒合理理解的 "數目"：題目本身不能算是一個解答。答案的運算式不能重複，但是可以有相同結果。如 $0+6= 6$ 和 $6+0= 6$ 算是兩個解答。

實例 1： 4

即: (3+5= 8, 3+6= 9, 6-0= 6, 9-0= 9) <-此行不用輸出

實例 2： 4

即: (0+6= 6, 0+9= 9, 6+0= 6, 9+0= 9) <-此行不用輸出

實例 3： 0

前言

請給定教室內有 N 位學生，並指定學生間 N 組友誼關係的可能性。(友誼關係: 是朋友或不是朋友)

每個學生的編號從 1 到 N ，且被編號 i 的學生稱為 i 。

在這間教室的學生都是利用他們的連繫網路來傳遞資訊。他們傳遞資訊的規則如下:

1. 首先，學生 1 會先接收到訊息。這個階段，除學生 1 之外沒有人知道關於這則訊息內容。
2. 接收到訊息的人會從友誼關係中未收過訊息的人中選擇一個人，並將訊息傳遞給他。
3. 當有友誼關係的人都知道訊息內容後，則結束傳遞。

輸出值為傳遞資訊人數的期望值。

誤差值可接受範圍為絕對誤差或相對誤差不超過 10^{-3} 。

輸入

輸入的標準格式如下:

```

N
C11  C12  C13  ...  C1N
C21  C22  C23  ...  C2N
:
CN1  CN2  CN3  ...  CNN

```

1. 第一個字串，讓使用者指定在教室內人數 $N(1 \leq N \leq 14)$ ，此為整數。
2. 第二個字串到第 N 個字串，讓使用者指定 N 組可能的友誼關係。
3. c_{ij} 表示整數 i 和 j 關係:
 - 如 c_{ij} 是 Y ，則學生 i 和學生 j 是朋友。
 - 如 c_{ij} 是 N ，則學生 i 和學生 j 不是朋友。

輸出

輸出值為傳遞資訊人數的期望值。

誤差可被接受範圍為，絕對誤差或相對誤差不超過 10^{-3} 。(小數點後三位)
此外，請在輸出後端插入換行的符號。

輸入例子 #1

```
4
NYYY
YNNN
YNNY
YNYN
```

輸出例子 #1

```
2.666
```

輸入例子 #2

```
5
NNNNN
NNYYY
NYNYN
NYNYN
NYNYN
NYNYN
```

輸出例子 #2

```
1.0
```

輸入例子 #3

```
10
NNNNYNYNNY
NNYYNNYNYN
NYNYYYNYYN
NYNYYYNYNY
YNYNYYYNN
NNYYNYYYN
YNYYYNYNYN
NNYNYYYNYN
NYYYNYYYNY
YNYNNNNYN
```

輸出例子 #3

```
9.501
```

前言

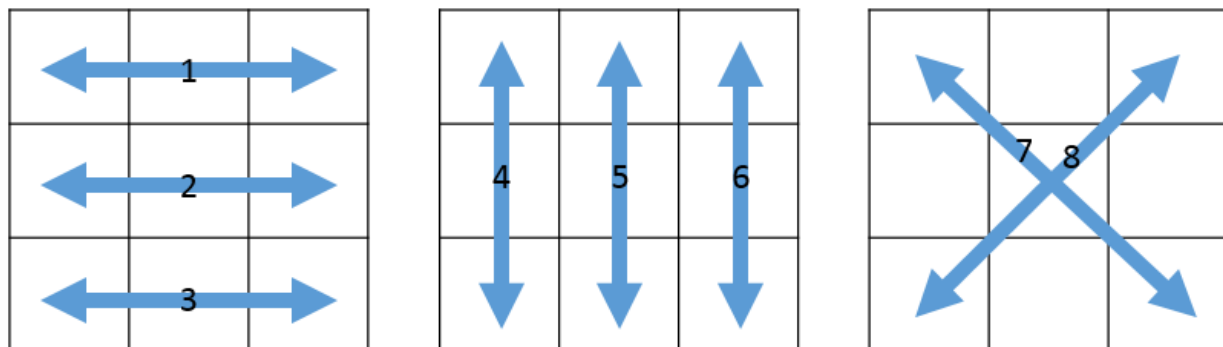
圖靈 (Alan Mathison Turing) 是計算機科學之父，同時也是人工智慧之父，為人類文明創造出無比的貢獻，近來在人工智慧領域有 DeepMind AlphaGo 首次打敗世界圍棋冠軍，明顯地，利用它將加速人類文明和生活。

在人工智慧的棋盤程式中，一般它會將各種可以走的步數列出來，然後透過一個函數（審局函數）將棋盤的狀態（輸贏）求出來，這個值關鍵地影響人工智慧中決策樹的樹枝修剪（Min-Max, Alpha-beta, Monte Carlo, ...），展開每個樹枝節點所需要的時間和空間成本，乃至於棋力的好壞（如：AlphaGo 第四局的情況）。

本題要實作一個簡易的井字遊戲的審局函數，以下是一般井字遊戲的權重表。

3	2	3
2	5	2
3	2	3

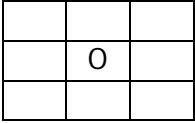
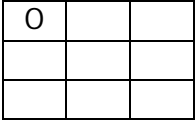
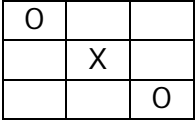
另外，井字遊戲中有橫向三條、直向三條、對角線二條，總共八條，加權計分方式如下：



	沒有佔子 (權重值)	佔一格	佔二格	三個都有
自己		50	200	1000
對手	0	-50	-200	-1000

如果八條線中的一條線雙方都有佔子，則該條線為 0 分。

棋盤資料：O 先 X 後 B 為空白，「自己」是指目前未下者，如 BBBBOBBBB，則「自己」代表 X。因審局函數有關效能，因此每個資料檔會有 100 – 2,000,000 筆棋局供計算。

	棋盤資料	八條線的計分（依序為）	合計
	BBBBOBBBB	8,-50,8,8,-50,8,-50,-50	-168
	OBBBBBBBB	-50,9,8,-50,9,8,-50,11	-105
	OBBBXBBBO	-50,50,-50,-50,50,-50,0,50	-50

輸入

資料檔名, 如：

A.txt

註：

- 第一行為權重表，第二行為加權表，接下來每一行為一局的資料，直到資料行沒有 9 個字元結束。
- [A.txt]
3,2,3,2,5,2,3,2,3
0,50,200,1000,0,-50,-200,-1000
BBBBOBBBB
OBBBBBBBB
BOBBBBBBB
OBBBXBBBB
OXBBBBBBB
OBBBXBBBO
OBBXBBBBB
OBXBXBBBO
OXBBXBBBO
OBXBXBOBO

輸出

為了驗證方便，請輸出所有棋局分數的加總。

-1100

註： 8,-50,8,8,-50,8,-50,-50 => -168

-50,9,8,-50,9,8,-50,11 => -105

-50,9,8,8,-50,8,11,11 => -45

50,-50,8,50,-50,8,0,-50 => -34

0,9,8,50,-50,8,50,11 => 86

-50,50,-50,-50,50,-50,0,50 => -50

-200,50,8,-50,0,8,0,50 => -134

0,-50,50,50,-50,0,0,-200 => -200

0,-50,50,50,-200,50,0,-50 => -150

0,50,-200,-200,50,0,0,0 => -300

-1100