

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。

CodeWars System 2.0

本次的比賽採用CodeWars System 2.0。使用手冊在您登錄進入系統後的桌面上。請您在比賽前先閱讀系統使用方式。

程式撰寫

本系統支援下列各種程式語言，請確定你會撰寫任何一種支援的語言。

Python 2.7.13

Python 3.6.0

GCC/G++ 6.3.0

Java OpenJDK 1.8.0_121

如果你是使用Java為程式語言，請注意Java class的命名必須為"codewars"，以下為範例說明。

```
public class codewars {  
    public static int gvar;  
    public static void say(String s) {  
        int x = 10;  
        System.out.print(s+x);  
    }  
    public static void main(String[] argv) {  
        float y = 0;  
        say("Hello, world\n");  
    }  
}
```

強烈地建議你在開始競賽之前，

先提交問題#0 (在下一頁上) 以確保你熟悉系統的使用方式。問題#0:送分題在

CodeWars System 2.0 上問題列表的最後一題(40 分)

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 HPE CodeWars 2018 Taipei!」。

輸入

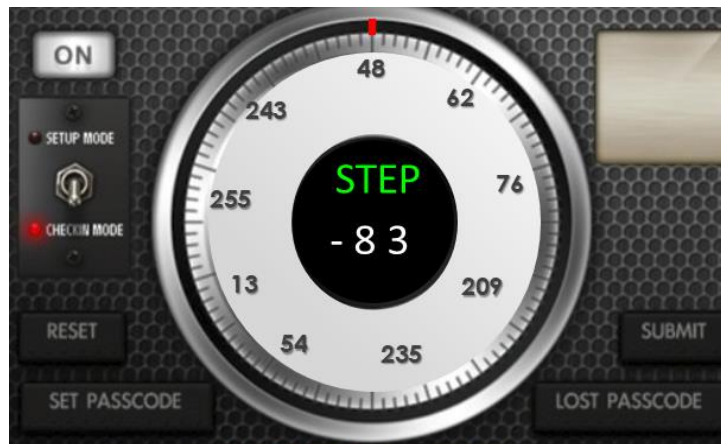
[這題沒有輸入。]

輸出

Hello HPE CodeWars 2018 Taipei!

前言

一座金庫的數字密碼序列會浮現在旋鈕旁邊，每次浮現出的數字序列長度及順序沒有固定，旋轉的方向及次數也沒有固定，根據每次旋轉提示由中心點向左轉或向右轉後，以中心點按照順時針順序方可得到真正的密碼序列 (請參考下方圖一)。



輸入

1. 先輸入數字序列的長度 N ， $5 \leq N \leq 20$ ，以此長度以亂數生成一數字序列(每個數字的範圍為 1~256)，數字間以空白區隔
2. 再來輸入移動序列的長度 M ， $3 \leq M \leq 10$ ，以此長度以亂數生成一移動序列(每個數字的範圍為 -100~100)，數字間以空白區隔，負數代表往逆時針方向轉, 正數代表往順時針方向轉

實例1：

```
5
23 77 129 14 232
3
-82 -72 12
```

實例2:

```
9
48 62 76 209 235 54 13 255 243
5
42 38 -68 -73 60
```

實例3:

14

172 55 22 204 114 117 255 239 232 141 232 242 211 118

10

-100 36 28 8 -88 -24 45 -90 -46 -91

輸出

實例1:

129 14 232 23 77

實例2:

62 76 209 235 54 13 255 243 48

實例3:

172 55 22 204 114 117 255 239 232 141 232 242 211 118

問題 2

天干與地支

6分

前言

在中國古代是以干支記錄年、月、日，干支是天干與地支的合稱，十天干分別是甲、乙、丙、丁、戊、己、庚、辛、壬、癸，十二地支是子、丑、寅、卯、辰、巳、午、未、申、酉、戌、亥，由兩者經一定的組合方式搭配成六十對，為一個周期，民國元年是壬子年，請聰明的妳/你設計一個程式可以將西元年轉換成民國年與中國的天干地支！

註: 民國元年為西元 1912 年。

輸入

輸入西元年份。

實例 1:

1975

輸出

程式必須輸出相符的民國與天干地支兩個元素，兩個元素之間以空白區隔。

實例1:

64 乙卯

問題 3

特務電碼

8分

前言

特務 T 在電碼中擷取了一段數字，只知道這段數字代表著一串英文字，而字母是依照：

A->1

B->2

C->3

...

Z->26

的方式替換且每個數字都是有對應的，但是不知道這些字母的長度與組合資訊，請寫個程式告訴特務 T 使用上列規則解碼這串數字有多少組合？

範例: 13-> 有可能代表 AC(1,3)或是 M(13) 所以有 2 種組合

輸入

傳入的字串 s 為非空字串且大於 0 之數字，字串最大為 10 個數字

實例 1: 101

輸出

輸出為組合數，數字組合必須為左到右。

如果無法組出該數字則輸出 0，如數字 1001 無法有合法組合，所以輸出為 0。

實例 1: 1

前言

將秘密訊息進行解碼。

每個收到的秘密訊息是透過兩步驟的加密：

1. 原始消息內的每個英文字母先轉為簡易 Pig Latin 語法

2. 用凱撒密碼加密成密文

簡易 Pig Latin 語法的規則是將每個字的第一個字母移到字尾，並加上

“ay”。例如：“hello”變成“ellohay”，而字串“hewlett packard”即為

“ewletthay ackardpay”

在轉為 Pig Latin 後，訊息會再進行凱撒加密，如圖一。在加密過程，每個字母會順著字母表往右

移動 n 個位置。例如，如果偏移量 n 為 3，原始訊息“hello”會變成“khood”，因為每個字母順著字

母表往右移動 3 個位置 ($h \rightarrow i \rightarrow j \rightarrow k$, $e \rightarrow f \rightarrow g \rightarrow h$, $l \rightarrow m \rightarrow n \rightarrow o$, $o \rightarrow p \rightarrow q \rightarrow r$)。凱撒加密的解碼方式就是往

相反方向（左邊）移動 n 位置。假設加密時使用了 $n=3$ ，“khzohww sdfndug”字串則會翻譯回成

“hewlett packard”，因為 $k \rightarrow j \rightarrow i \rightarrow h$, $h \rightarrow g \rightarrow f \rightarrow e$ 。

註：字母表的尾端會連接字母表的開頭 ($x \rightarrow y \rightarrow z \rightarrow a \rightarrow b \rightarrow c$)。

而由於本題目目的為解碼，需要將秘密訊息回復為原始訊息，也就是說需要先將凱撒密碼解密，

再利用 Pig Latin 翻譯回原始的英文字詞。輸入為凱撒加密的偏移量 n 和加密過的秘密訊息（可以

是超過一個字的字串），而輸出要是解碼過後的原始訊息。輸入資料的格式是偏移量 n 和每個字

由一個空格間隔。全部的字母皆為小寫。

輸入

實例 1:

```
1 fmpibz
```

實例 2:

```
3 hzohwwkdb dfndugsdb
```

輸出

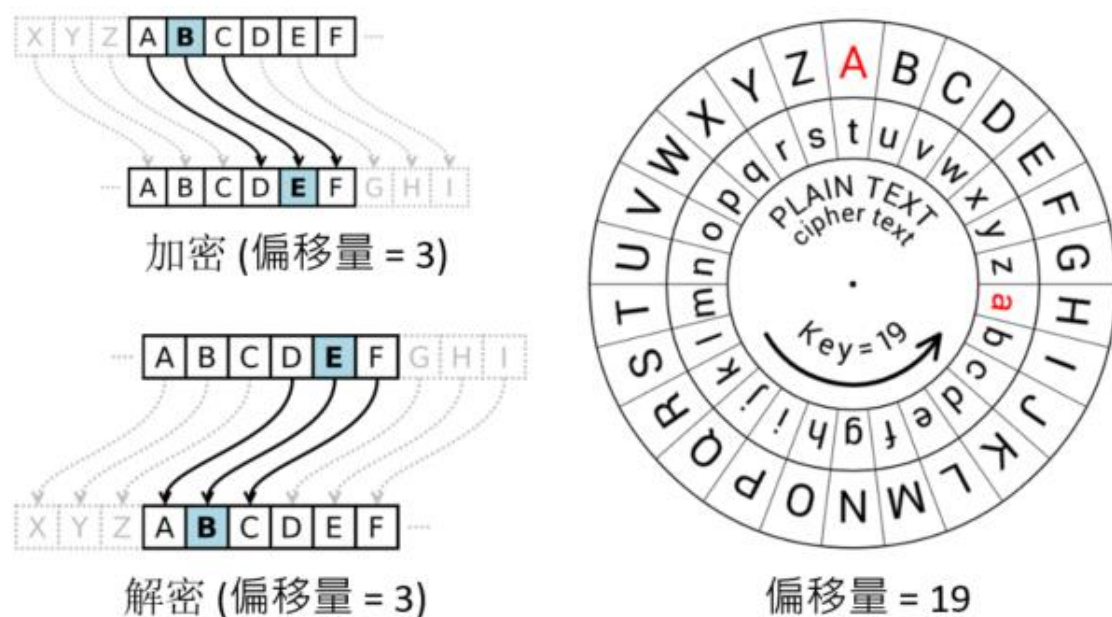
實例 1:

```
hello
```

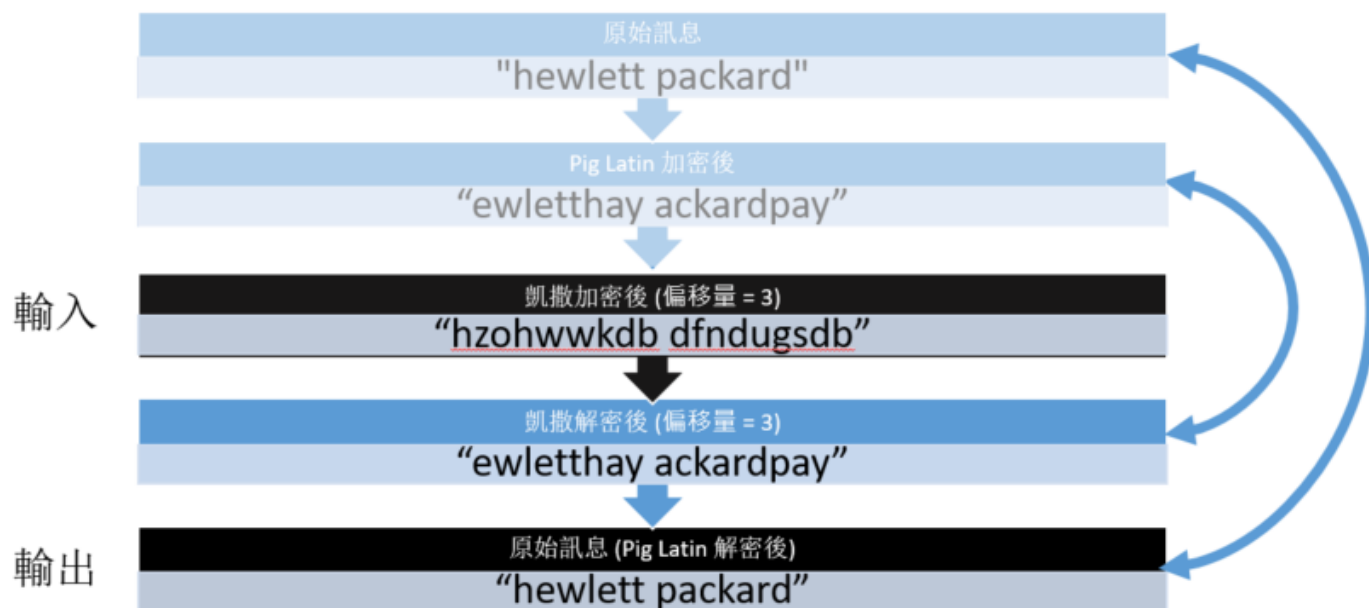
實例 2:

```
hewlett packard
```

圖一、凱撒加密範例



圖二、加密解密流程圖



前言

HPE 舉辦了一個促銷活動，參加者把有購物金額資訊的抽獎卷投到箱子裡。活動第一天早上箱子是空的，每天門市關閉之後，HPE 從箱子裡面取出購物金額最大和最小的兩張抽獎卷，前者對應的顧客將得到價值 $\max - \min$ （相減）的禮品。 \max 和 \min 分別是最大和最小的購物金額，然後把這兩張抽獎卷丟掉（**箱子裡其他抽獎卷仍然留在箱子裡**）。你的任務是計算整個活動 HPE 提供的禮品總價值。如果抽獎卷金額全部相等無法分出 \max 和 \min ，則禮品價值為 0，也是一樣丟掉兩張抽獎卷。

輸入

輸入一組資料，第一行為活動天數 n 。接下來的 n 行每行描述一天所收到的所有抽獎卷，其中第一個整數 k 是箱子內抽獎卷的個數 (0 代表當天沒有抽獎卷)。接下來 k 個整數是箱子內各抽獎卷的金額。輸入保證每天抽獎前，箱子裡至少還有兩張抽獎券。

實例 1：

```
2
2 1 2
2 1 2
```

實例 2:

```
5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2
```

輸出

活動期間提供的禮品總價值。

實例1:

```
2
```

實例 2:

```
19
```

圖例一

實例 2	每天箱子裡剩餘的抽獎卷	當天禮品價值
------	-------------	--------

問題 6

圖形判斷

12分

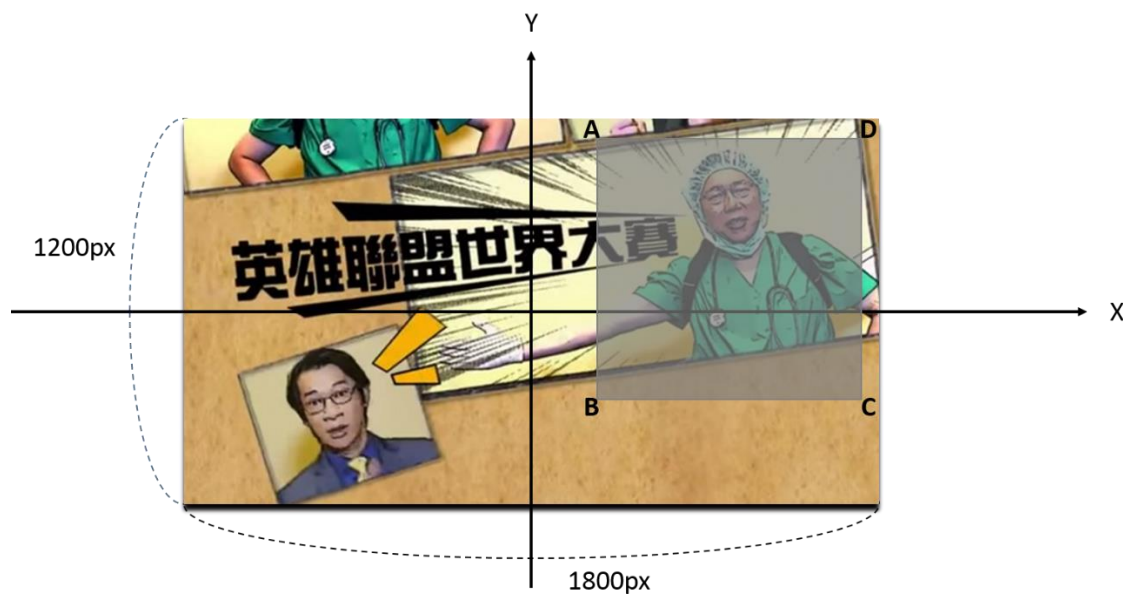
5		
3 1 2 3	1 2 3	2 (3-1)
2 1 1	2 1 1	1 (2-1)
4 10 5 5 1	1 10 5 5 1	9 (10-1)
0	5 5 1	4 (5-1)
1 2	5 2	3 (5-2)

禮品總價值: $2+1+9+4+3=19$

前言

Instagram 是現在最受歡迎的社群軟體之一，但你可知道在 2015 年之前 Instagram 有上傳影像和圖片大小的限制？沒錯，只能上傳「正方形」的檔案。當時為了力挺台灣參加「英雄聯盟 2015 世界大賽」，市長柯 P 與運動主播徐展元拍攝了一系列廣告。每張廣告的原始圖片大小皆為 $1800\text{px} \times 1200\text{px}$ ，若以圖片的正中間為座標原點，X 軸與 Y 軸和照片的高寬平行(如下圖)，座標單位為 px。廣告商必須從原始圖檔中裁切出一個平行於 X 軸的正方形用以放上 Instagram 做宣傳。任意給予四個座標點，請判斷這四個點是否為正方形。如果符合需求，請將原本沒有任何順序的四個座標點依照 A (左上)、B (左下)、C (右下)、D (右上) 的順序告知廣告商。

#2018CodeWar



輸入

輸入四個座標點，每一行輸入一點，一共四行。

先輸入 X 座標再輸入 Y 座標，並用一空白隔開。

X 為 -900 ~ 900 之間的整數。

Y 皆為 -600 ~ 600 之間的整數。

實例1:

```
0 0
10 10
0 10
10 0
```

實例2:

```
-20 5
-20 -5
-10 0
-30 0
```

輸出

程式必須輸出一個布林值(boolean)以表示判斷結果。如果符合需求，請再依照 A (左上)、B (左

下)、C (右下)、D (右上) 的順序輸出座標點；一行輸出一點，共四行。先輸出 X 再輸出 Y，中間用一空白隔開。

若不符合需求，輸出布林值即可。

實例1:

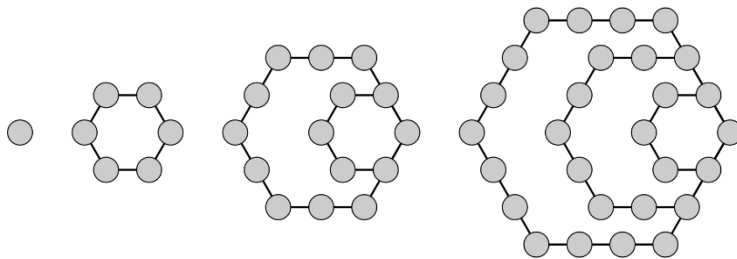
```
True
0 10
0 0
10 0
10 10
```

實例2:

```
False
```

前言

觀察以下圖一圖形的規律，不難發現，每一個圖形都是前一個圖形的外圍再加上一個正六邊形。



譬如第三個圖形，它就是第二個圖形的外面再圍上一個邊長為 3 個點的正六邊形；同理，第三個圖形的外邊再圍上一個邊長為 4 個點的正六邊形就變成第四個圖形。

利用這個規律，我們可以做出一系列邊長為 5、6、7、8 到任意個點的圖形。
計算每個圖形的點的數量，這個數量，我們稱做六邊形數。

譬如上面四個圖形，第一個圖形中，點的數量是 1；第二個圖形的數量是 6；第三個圖形的數量是 15；第四個圖形的數量是 28。

所有的六邊形數可以形成一個數列：1, 6, 15, 28, 45, 66, 91,

現在，給你任何一個正整數，你的程式需要找出最接近的六邊形數。如果同時有兩個六邊形數一樣接近的話，請輸入較小的那個數字。

舉例來說，輸入 8 這個數字的話，你的程式要輸出 6。因為 8 周圍的兩個六邊形數是 6 與 15，而 8 與 6 比較靠近。

輸入的正整數範圍: 最小為 1，最大為 4294967295。

輸入

每次的輸入是多個正整數，每個正整數一列。

實例1:

1
5

9
100

實例2:

493
9376
94666

輸出

針對每個輸入的正整數，輸出最接近的六邊形數。每個六邊形數一行。

實例1:

1
6
6
91

實例2:

496
9453
94830

問題 8

銅鑼燒特賣會！

15分

前言

HPE 點心店正舉辦銅鑼燒特賣會，胖虎老闆每次做完一批銅鑼燒後，會隨意的往檯面上的某幾個連續籃子放進一定數量的銅鑼燒，由顧客隨意挑選某一個籃子結帳。請幫幫哆啦 A 夢選出銅鑼燒最多的籃子，告訴他總共可以拿到幾個銅鑼燒！

$$3 < \text{籃子數} < 10^7$$

$$1 \leq \text{共幾批} \leq 2 \times 10^5$$

$$0 \leq \text{每次放進籃子的銅鑼燒數量} \leq 10^9$$

範例：

10 3 表示總共 10 個籃子，共計三批銅鑼燒。
 1 6 5 在編號 1 ~ 6 的籃子，各放進五個銅鑼燒。
 4 7 6 在編號 4 ~ 7 的籃子，各放進六個銅鑼燒。
 6 9 3 在編號 6 ~ 9 的籃子，各放進三個銅鑼燒。

	1	2	3	4	5	6	7	8	9	10		
[0	0	0	0	0	0	0	0	0	0]	十個空籃子
[5	5	5	5	5	5	0	0	0	0]	放進五個銅鑼燒
[5	5	5	11	11	11	6	0	0	0]	再放進六個銅鑼燒
[5	5	5	11	11	<u>14</u>	9	3	3	0]	再放進三個銅鑼燒

銅鑼燒最多的籃子是六號籃子，總共有 14 個銅鑼燒！

輸入

10 3
 1 6 5
 4 7 6
 6 9 3

輸出

14

前言

小明有一架無人機，可以做水平方向的飛行，直到遇上障礙物為止。以台北市內的大樓為例，若無人機由新光三越大樓起飛，無人機可以穿越或降落在延途中等高或較低的建築物。但若前方建物較高，例如台北 101，則無法穿越。

我們以1維空間來思考這個問題。假設由左至右總共有 N 棟大樓在飛行路徑上，而 i^{th} 大樓的高度為 h_i 。那麼小明可以把無人機的飛行路徑表示為 (i, j) ，意謂著無人機自 i^{th} 大樓起飛，而在 j^{th} 大樓降落。而路徑 (i, j) 為有效的條件即為，延途所有建物的高度， $h_i, h_{i+1}, h_{i+2}, \dots, h_{j-1}, h_j$ 皆小於等於 h_i 。無人機的起始位置於第一棟建物的頂樓，飛行方向僅為單向(向右飛行)，不會往回飛。第一棟建物尋找完可飛行路徑後，再由第二棟建築物起飛尋找可飛行路徑，依此類推至倒數第二棟建築物(最後一棟不算)。

輸入

第一行輸入為 N ，表示有多少建物，第二行為 N 個空白間隔的數字，表示建築物的高度。
 $1 \leq N \leq 3 \times 10^5$ ，而建物高度介於 $1 \sim 10^6$ 。

實例1:

6
3 2 1 2 3 3

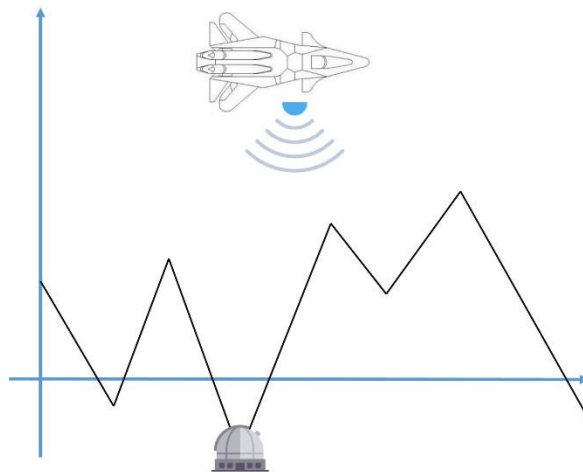
輸出

印出有效路徑的總數

實例1: 8

前言

復仇者聯盟為了解決巡防資源不足的問題，正在著手開發自動偵測破壞者基地的系統。此系統需要從光學雷達取得地形圖資料，並加以分析此資料，找出回傳資料串中最深的坑洞，此坑洞極有可能為破壞者的基地。



資料為一個非空元素，0 索引為基礎的 A 陣列，找出陣列 A 中坑洞(P, Q, R)最大深度。

坑洞(P, Q, R) 必須滿足下列條件:

- $0 \leq P < Q < R < N$ (陣列索引值)
- 陣列下降條件 $A[P], A[P+1], \dots, A[Q]$, 例如: $A[P] > A[P+1] > \dots > A[Q]$
- 陣列上升條件 $A[Q], A[Q+1], \dots, A[R]$, 例如: $A[Q] < A[Q+1] < \dots < A[R]$

坑洞(P, Q, R) 深度計算方式如下圖公式:

$$\text{Min}\{ A[P] - A[Q], A[R] - A[Q] \}$$

例如, 下列為 10 個元素的陣列 A :

$$A[0] = 0, A[1] = 1, A[2] = 3, A[3] = -2, A[4] = 0$$

$$A[5] = 1, A[6] = 0, A[7] = -3, A[8] = 2, A[9] = 3$$

坑洞(2, 3, 4) 為陣列 A 的其中一組坑洞，因為 A[2], A[3] 滿足下降條件 ($3 > -2$) 且 A[3], A[4] 也滿足上升條件 ($-2 < 0$)，而它的深度為 $\min(A[2]-A[3], A[4]-A[3]) = 2$ 。坑洞(2, 3, 5) 也是陣列 A 中的另一組坑洞，而它的深度為 3。坑洞(5, 7, 8) 也為陣列 A 中一組坑，其深度為 4。在陣列 A 中，坑深度最大為 4，故輸出答案為 4。

若陣列中，沒有存在任何一組坑洞的話，其輸出答案為 -1。

輸入

輸入一組資料，第一行為陣列長度 ($10 \leq \text{陣列長度} \leq 255$)，第二行為陣列內容。陣列數值介於 -100 到 100 之間。

實例1:

```
10
0 1 3 -2 0 1 0 -3 2 3
```

實例2:

```
11
1 2 3 4 5 4 3 2 1 0 0
```

輸出

程式必須輸出坑洞的最大深度。

實例1: 4

實例2: -1

前言

HPE 舉行麥克雞塊特賣會，這次特賣會顧客可以自由選擇購買多少個雞塊，而你的任務是使用不同的盒子去盛裝，但是為了不浪費紙盒，你必須要剛好把使用的每個紙盒裝滿；請依照給定的盒子種類與顧客購買的麥克雞塊數量找出最少盒裝組合的盒子數量。

舉例來說，顧客購買了 15 個麥克雞塊，而 HPE 這次使用的紙盒有 3 塊、5 塊和 7 塊盒裝三種，這時候你可以選擇使用 5 個 3 塊盒裝、3 個 5 塊盒裝或者 3 塊 5 塊 7 塊盒裝各使用 1 個等這三種盛裝方法，由此可知 15 塊麥克雞塊最少需要 3 個盒子。

若找不出任何組合則回傳 0。

輸入

輸入的第一行代表的是盒子種類的個數，第二行代表的是各種盒子盛裝的大小，第三行是顧客購買的數量。

實例 1:

3
3 7 5
15

實例 2:

4
7 2 6 11
32

實例 3:

2
3 5
7

輸出

最少使用的盒子數量。

實例1: 3

實例2: 4

實例3: 0

前言

給定區域 $M(x\text{方向}) \times N(y\text{方向})$ ，兩隻速度相同且同時出發的機器人R1、R2在區域中行走。

機器人R1:

帶有 $F1$ 的油量，每走一步消耗一單位的油。R1從 $(X1, Y1)$ 開始，重複 $N1$ 步向北後走 $E1$ 步向東，再走 $N1$ 步向北， $E1$ 步向東，如此重複直到油耗完。

機器人R2:

帶有 $F2$ 的油量，每走一步消耗一單位的油。R2從 $(X2, Y2)$ 開始，重複 $E2$ 步向東後走 $N2$ 步向北，再走 $E2$ 步向東， $N2$ 步向北，如此重複直到油耗完。

區域邊界有傳送門會將機器人送到對面。如區域大小為 4×5 ，當機器人在 $(3,3)$ 向東走後將被傳送到 $(0,3)$ ，同時也消耗一單位的移動量和一單位的油。

請判斷兩機器人是否會碰撞。

輸入

只有一行，包含 $M, N, X1, Y1, E1, N1, F1, X2, Y2, E2, N2, F2$ ，以空白分隔。

- $0 < N, M < 10000$
- $0 \leq X1, X2 < M$
- $0 \leq Y1, Y2 < N$
- $(X1, Y1) \neq (X2, Y2)$
- $0 < N1, E1, N2, E2$
- $0 \leq F1, F2 \leq 1000000$

實例 1:

7 6 2 0 9 2 100 3 5 2 7 100

實例 1:

7 6 2 0 9 2 6 3 5 2 7 0

輸出

印出兩者在第幾個時間點 T 相撞，若不會相撞，則輸出 -1。

實例 1: 5

實例 1: -1

前言

復仇者聯盟的蟻人(A) 喜歡吃餅乾，他會定期拜訪他的朋友們：黑寡婦(B)、美國隊長(C)、德克斯(D)、格魯特(G)、浩克(H)、鋼鐵人(I)、洛基(L)，並帶給他們一些餅乾。

當蟻人(A) 拜訪鋼鐵人(I)時，會帶給他 (p) 個餅乾；拜訪浩克(H)時，會帶給他 (q) 個餅乾。如果他拜訪格魯特(G)，則需要先去鋼鐵人(I)或浩克(H)的家。也就是說，如果他選擇先去鋼鐵人(I)的家，會帶 ($p + u$) 個餅乾；先去浩克(H)的家，會帶 ($q + v$) 個餅乾。

圖一為蟻人(A)拜訪每間朋友所需準備的餅乾數量，請幫助蟻人(A) 計算拜訪每位朋友，所需準備的最少餅乾數。

範例一請參考圖二，範例二請參考圖三

輸入

餅乾個數 $o, p, q, r, s, t, u, v, w, x, y, z$

實例 1: 4 8 3 6 5 4 7 6 9 3 6 4

實例 2: 2 7 5 5 4 6 4 7 8 1 7 6

輸出

拜訪每位朋友 B, C, D, G, H, I, L 所需準備的最少餅乾數

實例 1: 4 7 13 9 3 8 9

實例 2: 2 7 10 11 5 7 6

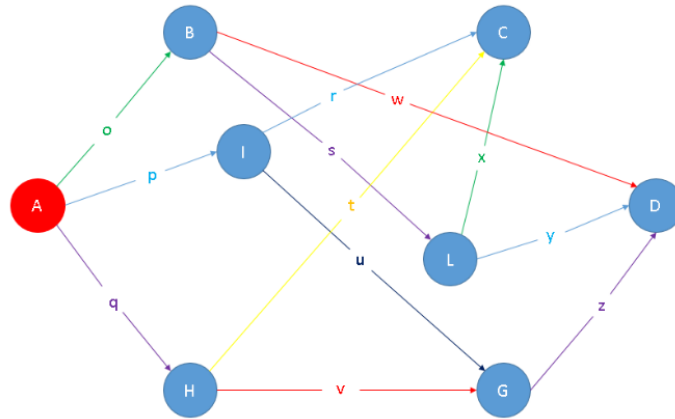


Figure 1

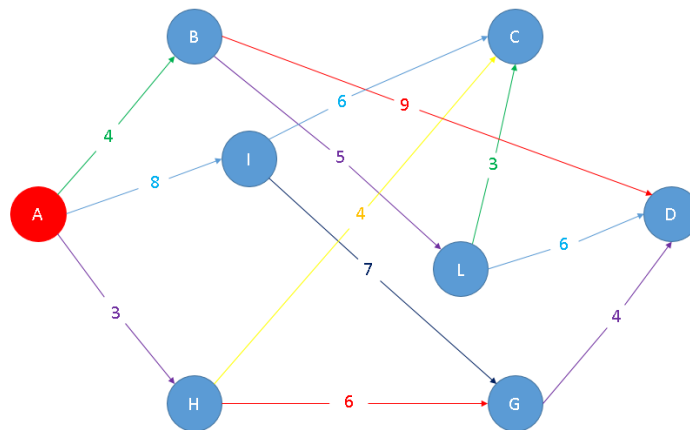


Figure 2

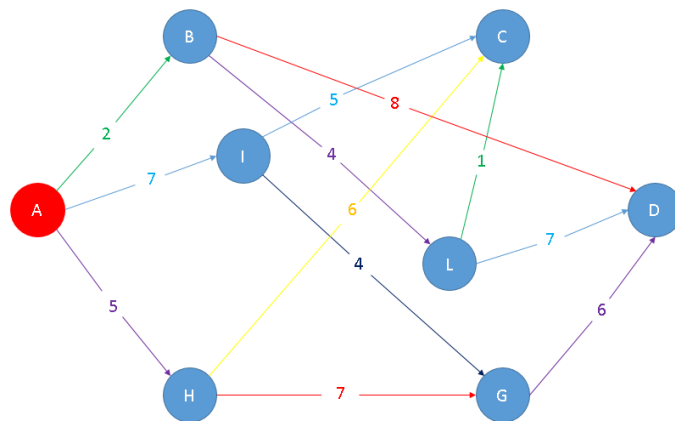


Figure 3

前言

HPE 點心店全新鹹蛋苦瓜銅鑼燒上市後，引爆排隊熱潮，排隊時大家都會拿到從 1 開始逐一發放的號碼牌，正巧 TWICE 子瑜也來排隊，隊伍變得一團混亂。胖虎請哆啦 A 夢幫忙用最少的交換次數，把隊伍恢復原狀，就免費送哆啦 A 夢全新鹹蛋苦瓜銅鑼燒。

請幫幫多拉 A 夢讓他可以獲得最好吃的鹹蛋苦瓜銅鑼燒吧！

$1 \leq \text{排隊總人數} \leq 10^5$

輸入範例：

輸入：

```
5                ←總共 5 人
3 4 5 1 2       ←隊伍目前的號次

3    4    5    1    2    ←原本的隊伍
1   4    5    3   2    ← 1 和 3 交換
1    2    5    3    4    ← 2 和 4 交換
1    2    3   5   4    ← 3 和 5 交換
1    2    3    4   5   ← 4 和 5 交換，耶！
```

輸出

4

輸入：

```
7                ←總共 7 人
6 3 5 2 4 1 7   ←隊伍目前的號次

6    3    5    2    4    1    7    ←原本的隊伍
1   3    5    2    4    6   7    ← 1 和 6 交換
1    2   5    3   4    6    7    ← 2 和 3 交換
1    2    3   5   4    6    7    ← 3 和 5 交換
1    2    3    4   5   6    7    ← 4 和 5 交換，耶！
```

輸出

4

輸入

實例1:

7

7 1 3 2 4 5 6

輸出

實例 1: 5

前言

Mr. Taco 想去拜訪住在深海大鳳梨裡的海綿寶寶。攤開地圖後他發現從家裡到大鳳梨的路上布滿了各種不同的珊瑚礁。珊瑚礁之間藏著玳瑁、海鰻或鯊魚，這三類海洋的居民對 Mr. Taco 的態度都不是很友善，甚至會主動攻擊 Mr. Taco，因此 Mr. Taco 想盡量避開他們趕快到達大鳳梨。根據過往的經驗，Mr. Taco 將玳瑁、海鰻和鯊魚的危險程度分別做了標記，數字越大，危險程度越高。現在 Mr. Taco 將地圖分享給你，請幫忙找出一條從家裡到大鳳梨，危險程度最低的一條路線！

輸入

第一行輸入 n 、 m 、 i 、 j 、 k ，以空格分開，代表地圖有 n 座珊瑚礁，礁與礁之間共有 m 條路。其中第 1 號珊瑚礁視為 Mr. Taco 家，第 n 號珊瑚礁視為大鳳梨。 i 、 j 、 k 分別為玳瑁、海鰻和鯊魚的危險程度。

第二行至第 $m+1$ 行，每行各有 x 、 y 、 a 、 b 、 c 三個數字，以空格分開。代表第 x 座珊瑚礁與第 y 座珊瑚礁之間有 a 隻玳瑁、 b 隻海鰻和 c 隻鯊魚。每座珊瑚礁之間的危險程度視為玳瑁、海鰻和鯊魚之危險程度的總和。

實例1:

```
5 6 1 2 3
1 2 2 1 0
1 3 1 0 0
2 3 1 1 0
3 4 0 0 1
3 5 0 1 1
4 5 1 1 0
```

實例2:

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

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

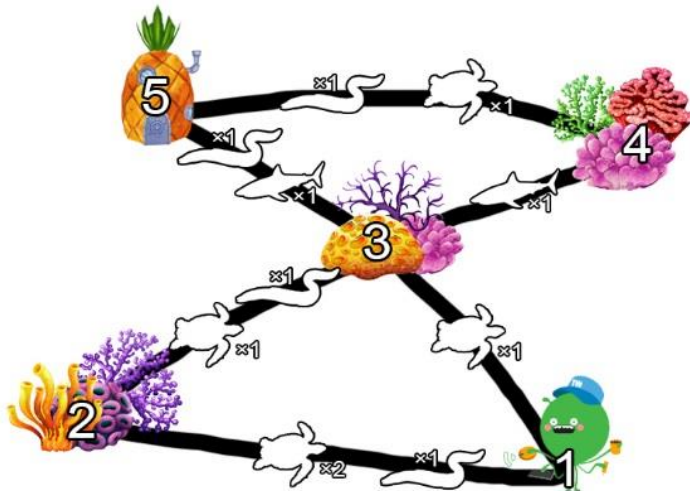
輸出

程式必須輸出由第 1 號珊瑚礁(Mr. Taco 家) 抵達第 n 座珊瑚礁 (大鳳梨)，途中需要依序經過的珊瑚礁號碼，且此路途必須是珊瑚礁危險程度總和最低的路途。答案已保證唯一，不會有兩條以上的路途。珊瑚礁號碼的數字之間須以空格分開。

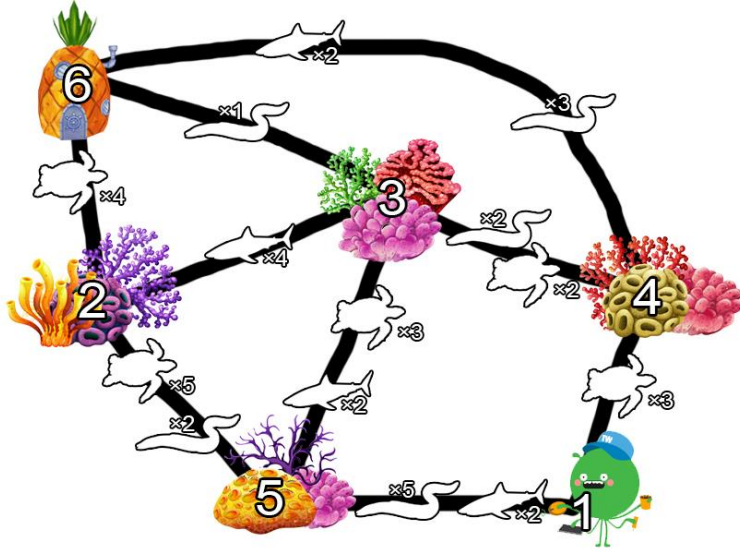
實例1: 1 3 5

實例2: 1 4 3 6

圖例一



圖例二



前言

薩諾斯要蒐集到六顆寶石才有辦法實現他的心願：消滅掉一半的人來維持全宇宙的和平，目前薩諾斯只差最後一顆 時間寶石，不過一直不知道這個寶石的下落，這時候看到一段文字

這顆神奇的時間寶石的位置有可能在地球上，千年以前一場變異讓整個地球充滿了神秘的力量，這力量連最有智慧的博士都沒有辦法理解，時間到底是什麼呢？我們可不可能有時間或機會證明我們可以回到過去或是前進未來，這一顆寶石可能就是唯一的解藥。

薩諾斯發現這篇文字透漏了「奇異博士有時間寶石」。

輸入

輸入有兩組，

1. 搜尋目標字串，內含兩種資料
 - a. 第一個數字為陣列長度，
 - b. 第二個資料即為目標搜尋字串一維陣列；
2. 搜尋來源，內含兩種資料
 - a. 第一個資料包含兩個數字，第一個數字為 行 (row) 數量，第二個數字為列 (column) 數量
 - b. 第二個資料即為來源陣列，以一為陣列呈現，需先按照上面的數字轉換成二維陣列

假設第二組輸入為

```
3 2
A B C D E F
```

需先轉換成二維陣列再進行搜尋

```
[
    [A, B],
    [C, D],
    [E, F],
]
```

轉換為二維陣列之後，在此二維陣列當中尋找是否存在連續目標字串，只需考慮上下左右方向即可。若存在，則回傳 true，反之則回傳 false。

實例 1:

```
5
A P P L E
3 8
A B C D E F G H P P T D D P A S Z L A E L P A S
```

經轉換過後的二維陣列為

```
[
  [A, B, C, D, E, F, G, H],
  [P, P, T, D, D, P, A, S],
  [Z, L, A, E, L, P, A, S],
]
```

可於二維陣列當中找到第一組輸入的連續字元 **APPLE** (粗體)

實例 2:

```
5
A P P L E
3 8
A B C D E F G H P P T D G D A S Z L A D G S A S
```

經轉換過後為的二維陣列為

```
[
  [A, B, C, D, E, F, G, H],
  [P, P, T, D, G, D, A, S],
  [Z, L, A, D, G, S, A, S],
]
```

無法於二維陣列中找到第一組輸入的連續字元 **APPLE**

輸出

輸出 "True" or "False"

實例1: True

實例2: False

前言

鋼鐵人出任務，這次的場景是 80 年代的經典機關倉庫番。為了節約能源，神盾局希望能以箱子最小的移動數來完成每一個關卡。

什麼是倉庫番?鋼鐵人從 Wiki 下載資料查詢如下

倉庫番的規則[\[編輯\]](#)

第一個《倉庫番》的遊戲規則，則是扮演工人的玩家，以「推」的方式，推動箱子。可以在沒有阻礙物（如牆壁等的阻礙物）的情況下，向左、右、上、下的地方移動。將箱子移動到指定點，達到指定數量，即可過關。

但玩家移動箱子，有下列條件要注意：

- 推到牆壁的箱子，玩家就不可以背對牆壁，把箱子推回到空處。即箱子只能以「被推」的方式被移動，不是以「被拉」的方式被移動。但如果玩家推至牆壁，垂直牆壁的兩側沒有阻礙物，則玩家可以朝這兩個不同的方向推移箱子。
- 一旦箱子被移動到角落，玩家沒有任何方法再移動這個被推到角落的箱子。

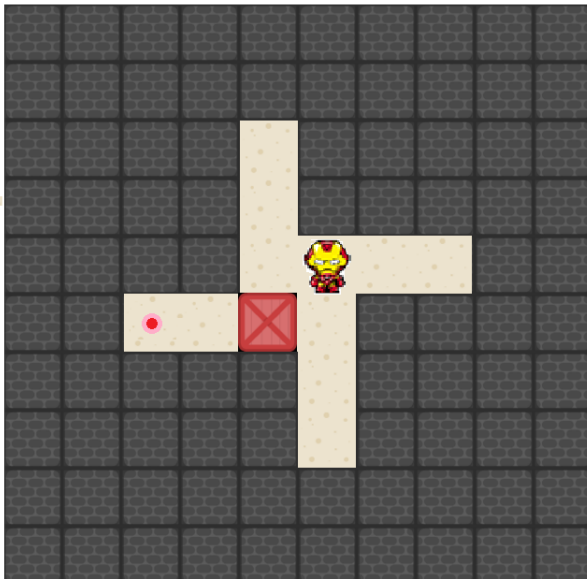


示意圖1

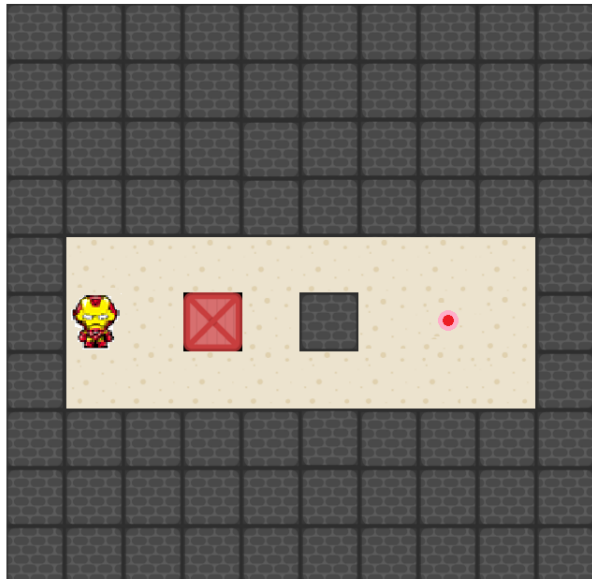


示意圖2

題目輸入規則

- 將地圖 (見示意圖 1&2) 固定用[10][10]的二維方式表達 (見實例 1&2)
- 元素 0 表示允許鋼鐵人跟箱子可經過/停留之空地
- 元素 1 表示不允許鋼鐵人和箱子經過/停留之牆壁
- 元素 2 表示鋼鐵人本尊所在之空地
- 元素 3 表示箱子所在之空地
- 元素 4 表示預計推動箱子最終到達的目標空地

答案輸出規則

- 箱子可推達目標時輸出箱子最小移動次數；否則輸出-1

#我們的三大安心保證

#保證只有一個箱子

#保證鋼鐵人不在牆上也不在箱子上，但一定在地圖上

#保證牆沒有破洞，箱子不會推出地圖，鋼鐵人也不能溜出去

#鋼鐵人沒有點穿牆/跳箱技能點數

#其實比較想知道鋼鐵人最少需要的移動步數

#節能減碳愛地球，救救北極熊

輸入

實例1:

```
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 0 2 0 0 1 1
1 1 4 0 3 0 1 1 1 1
1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

實例2:

```
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

```
1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 1
1 2 0 3 0 1 0 4 0 1
1 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

輸出

實例1:

2

實例2:

-1

前言

香港警隊為了攻破由韓琛為首的販毒集團，派出了陳永仁等數名臥底潛入韓琛的黑幫作為長期的臥底。陳永仁等人果然不負所托，在與重案組警官黃志誠的密切合作之下，終於攻破了韓琛的販毒集團。

案件結束之後，陳永仁等臥底警員就回到警隊報到，想要恢復警察身分。很不幸的，黃志誠警官早在與臥底的密會行動中不幸喪命，而黃志誠警官獨有的臥底名冊系統的主程式也在行動中不幸損毀。

這下糟糕了。沒了主程式，警隊就沒辦法辨認這些自稱是臥底警員的人員到底是真臥底還是假臥底。

為了幫助真臥底警員重回警隊，並且揪出假臥底，我們希望你的幫助，幫忙重建臥底名冊系統的主程式。

警隊目前知道，在派出每位臥底之前，都有給每位臥底一個祕密的數字 x ，這個祕密的數字只存在於該名臥底跟黃志誠警官心中，而且每位臥底的 x 數字是不同的。更重要的是， x 只能放在心

裡，不能說出來，不能寫在紙上，不能貼 facebook，更不能告訴你喜歡的女生。總之， x 只能存在心中。

現在黃志誠警官去世了，這些回歸的人員每個人都說自己有 x ，讓人無法分辨真假。

而你的臥底名冊系統的主程式，就是要幫助警隊在這些人員不能透漏 x 的情況下，揪出假臥底。

當然這些資訊是不夠幫助你重建臥底名冊系統的。我們採訪到梁警司(黃志誠的上司)，他私下透漏，該系統對每位人員的輸入是以下的格式: (數字部分只是範例，範例數字可能是假臥底輸入的錯誤資料)

```
p=524287
g=648284036501
y=124197
C=498634
r=28984
C=118791
R=10026
```

... (一連串的 C/r 或 C/R)

END

其中:

1. p 是質數
2. g 是一正整數，範圍介於 10^6 與 10^{12} 之間
3. p 與 g 是系統與人員在開始驗證前就講好的。
4. 人員會輸入 y ，其中， $y = g^x \bmod p$
5. 接下來人員會輸入一連串的
C=
r=
或是
C=
R=
6. C、r、R 的產生是這樣的：每一輪，人員都會隨機產生一個正整數 r ， $\frac{p}{2} \leq r \leq p-1$ 。接著，人員丟硬幣決定，如果是正面的話，這輪要輸入
C=...
r=...
如果是反面的話，這輪要輸入
C=...
R=...

C 與 R 的規則只有真臥底才知道，假臥底是不知道的。

規則是：

$$C = g^r \bmod p$$
$$R = (x + r) \bmod (p - 1)$$

7. 在若干輪的 C、r、R 輸入完畢後，你的系統要判定，該名人員有沒有可能是假臥底。
8. 在所有人的輸入都完成後，會以 END 字串結束。

Input

系統一次可以同時驗證好幾個人，每個人輸入的資料都是從 p 開始。譬如，以下是兩位人員的輸入：

p=8629

g=851589060216

y=2929

C=3316

r=5617

C=8405

R=7635

C=8295

r=4783

p=9973

g=412326166501

y=3253

C=8000

r=7821

C=8750

r=9316

END

Example 1:

p=10007

g=427829658531

y=559

C=5564

r=7743

C=2897

r=9462

END

Example 2:


```
p=20357
g=825878145731
y=14112
C=3339
r=11785
C=14654
R=1802
p=20963
g=104947618901
y=8007
C=2083
R=1742
C=936
r=15706
C=908
r=16691
END
```

Output

針對每一位人員的判定輸出 1 或是 0。如果確定是假臥底的話，輸出 0。如果輸入資料看起來都很合理，找不出破綻，似乎是真臥底，就輸出 1。每個數字獨立一行。

Example 1:

1

Example 2:

0

0

問題 19

神盾局偵測系統

36分

前言

宇宙中每個生命都有個靈魂水晶儲存著靈魂的能量，每一個靈魂在每一世結束要轉生時，可以用自由意志來決定下一生要生為哪個種族。例如：地球人如要想在下一世轉生為阿斯嘉人，必須花費 3 單位的靈魂水晶能量。而地球人如果下一世投生為永恆族，可以賺取靈魂水晶 2 個單位能量。條件是轉世後靈魂水晶能量要 > 0 。

第 N 世	第 N+1 世					
	阿斯嘉	天神族	格魯特	地球人	冰巨人	永恆族
阿斯嘉	-2	+1	-2	+3	+2	+1
天神族	-5	-2	+5	+2	+5	-4
格魯特	-5	-5	0	+5	-1	-1
地球人	-3	-2	-5	0	+1	+2
冰巨人	+5	-3	-2	-1	-3	+1
永恆族	-5	-4	-5	+2	+4	-1

第一世，也就是現世，我們編號為第 0 世。每 2 次轉生，靈魂水晶會有些損壞，需要花費能量來修補。

例如，此生為第 $N=0$ 生，下一生為第 $N=1$ 生，再下一生為第 $N=2$ 生。在第 $N=2$ 生時，因為已經轉生 2 次，靈魂水晶會有損壞，需花費

$$P = \left| \text{目前靈魂水晶能量值} \bmod 10 \right| + N \text{ (注意要取絕對值) 個單位能量來修補。}$$

例如這次轉生需要修補靈魂水晶，為第 4 次轉生，目前的靈魂水晶能量是 18，需花費 $18 \bmod 10 + 4 = 8 + 4 = 12$ ，要花費 12 個能量單位來修補水晶。

如果轉生後靈魂水晶能量 < 0 ，來自另一個未知宇宙的力量，會將靈魂水晶的能量加上 $N*N \bmod 30$ 的能量。如此，完成一個轉世的循環。

例如上一世是地球人，靈魂水晶能量為 6，這一世轉生為格魯特，靈魂水晶能量要消耗 5 個單位能量 (請看轉生規則表格)。如果這生為第 10 世，需修補，修補將會消耗

$$P = \left| (6-5) \bmod 10 \right| + 10 = \left| 1 \bmod 10 \right| + 10 = 11$$
，11 個能量單位，那麼這一世的靈魂水晶能量 = 6 上一世的靈魂水晶能量 -5 地球人轉生為格魯特 (查表) - 11 修補靈魂水晶 = -10，符合未知宇宙力量的補充條件 (轉生後靈魂水晶能量 < 0)，所以會補 $N*N \bmod 30 = 10 * 10 \bmod 30 = 100 \bmod 30 = 10$

最後總靈魂水晶能量 = -10 + 10 = 0。

當一個生命的靈魂水晶在修補損壞「之後」能量值 < 0 ，此靈魂就會化為暗黑物質，不再轉生。但是能量值 = 0 時，就產生奇異點，並建立通往另一個邪惡宇宙的蟲洞，邪惡生物會經過蟲洞來到我們的宇宙進行破壞。所以神盾局要建立一系統，可以偵測宇宙中每一個生命每一最快在第幾次轉生會讓靈魂水晶的能量 = 0，以便及早消滅蟲洞，防止邪惡生物的入侵。你的任務便是設計出此系統。

宇宙各種族編碼如下

阿斯嘉	天神族	格魯特	地球人	冰巨人	永恆族
0	1	2	3	4	5

輸入

輸入為兩個數字，第一個為種族編碼。第二個為目前靈魂水晶能量，兩數字間以空白區隔。

實例一:有一阿斯嘉人(查上表，編號0)，目前靈魂水晶能量為 2

0 2

實例二:有一冰巨人(查上表，編號4)，目前靈魂水晶能量為 45

4 45

輸出

實例一:

1 (轉生為阿斯嘉或是格魯特，靈魂水晶能量-2，等於 0)

實例二:

6 (第0世冰巨人→第1世阿斯嘉→第2世格魯特→第3世天神族→第4世阿斯嘉→第5世格魯特→第6世阿斯嘉)

補充

mod 為「取餘數」，例如：

$5 \div 3 = 1 \dots 2$ ，所以 $5 \bmod 3 = 2$

$49 \div 7 = 7 \dots 0$ ，所以 $49 \bmod 7 = 0$

$66 \div 5 = 13 \dots 1$ ，所以 $66 \bmod 5 = 1$

C / Java / Python 取餘數運算子都是 %

$5 \% 3 = 2$

$49 \% 7 = 0$

$66 \% 5 = 1$