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。

問題 0
送分題
21 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 HP CodeWars 2014 Taipei!」。

輸入

[這題沒有輸入。]

輸出

```
Hello HP CodeWars 2014 Taipei!
```

問題 1
檔案內容處理
3 分

前言

讀取一個檔案的內容並且轉換成一個特定格式。

讀入的檔案名稱為 **input.txt**。請以檔案內容裡的非英文字母(非 a~z 與非 A~Z)來分開檔案內容，輸出所有英文字母(a~z 與 A~Z)，並於分開之處輸出一個換行字元。以下為檔案內容範例：

```
Hewlett-Packard is the best company ever!
  **I love **  ******** working at HP ()_@”
```

輸入

[本題無須輸入]

輸出

```
Hewlett
Packard
is
the
best
company
ever
I
love
working
at
HP
```

問題 2
給力的瓦特計
3 分

前言

歐姆定律說明了**電壓**、**電流**及**電阻**之間的關係。

依據歐姆定律：

電壓 = 電流乘以電阻	V = I * R
電流 = 電壓除以電阻	I = V / R
電阻 = 電壓除以電流	R = V / I

瓦特定律說明了電壓、電流跟功率之間的關係。

依據瓦特定律：

功率(瓦特) = 電壓(伏特)乘以電流(安培)	P = V * I
功率(瓦特) = 電流(安培)平方乘以電阻(歐姆)	$P = I^2 * R$

台灣電力公司提供台灣地區住家每戶110V的電壓。每戶的電費是用「度」〔千瓦/小時〕來計量的，此瓦特計會依據輸入的電阻及使用分鐘數來計算一共用了幾度。

輸入

輸入包括兩組以空白分隔的數字，第一個值代表電阻值〔歐姆〕最多到小數點後一位，第二個值代表使用分鐘數(正整數)。實際案例請參考以下。

實例1： 10 40
實例2： 6.5 62
實例3： 3.8 38

輸出

程式必須印出使用了幾度電，四捨五入到小數點以下三位。

實例1： 0.807
實例2： 1.924
實例3： 2.017

問題 3
國道計程收費
3 分

前言

台灣國道改變以前的收費方式，改成國道計程收費。
國道計程收費方案，每天固定免費里程20公里，超出部分在200公里內、小型車每公里收費1.2元，200公里以上每公里收費0.9元。

請設計一簡易計算國道收費程式。
只使用下面條件，簡單的計算小型車在國道上行駛距離與費用。

1. 只考慮小型車。
2. 單次計算，每次計算固定免費里程20公里。
3. 21公里(含)以上到200公里(含)以下，每公里收費1.2元。
4. 201公里(含)以上每公里收費0.9元。

輸入

車輛在國道上行駛距離(輸入為正整數，單位：公里)

範例1：
```
18
```

範例2：
```
240
```

範例3：
```
200
```

輸出

國道收費費用(輸出為整數，小數點後四捨五入，單位：元)

範例1：
```
0
```

範例2：
```
252
```

範例3：
```
216
```

問題 4
三角形的頂點
4 分

前言

三角形的其中一個基本性質是任意兩邊和必須大於第三邊。若這個三角形是直角三角形則必須滿足畢氏定理，也就是某兩邊的平方合會等於第三邊(長邊)的平方，符合這個定理的三角形便是所謂的直角三角形。

一位數學老師為了測驗學生是否了解以上的性質，便出了幾題三角形的題目，讓學生算出三角形邊長以外，並讓學生熟悉畢氏定理。所以數學老師必須給定三角形的三個頂點讓學生運算。然而數學老師根據上面的性質找出了符合性質的邊長以外並確認這些邊長構成的三角形在平面座標系裡，至少可以找到一組頂點坐標都是整數。現在請你幫忙找出一組三角形的頂點坐標均是整數。你可能需要用到兩點距離公式如下：

若 $A(x_1, y_1)$ 且 $B(x_2, y_2)$，則 $\overline{AB} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

程式輸入為直角三角形的短邊長與斜邊長。程式輸出需為三角形的三個頂點且必須滿足以下條件：

1. 三角形頂點座標必須是正整數或 0。
2. 三角形的所有頂點的 x 座標和是最小的。
3. 頂點輸出順序需以 x 座標大小排列，小的排在前面，若有兩個頂點 x 座標相同則以 y 座標大小排列，小的排在前面。

輸入

```
3,5
```

輸出

```
(0,0),(0,4),(3,0)
```

問題 5
點餐系統
4 分

前言

瓊斯開了一家早餐店，以下是他所販售東西的價目表:

A1 原味蛋餅	18	A2 鮪魚蛋餅	25
B1 豬肉三明治	25	B2 牛肉三明治	30
C1 香雞蛋漢堡	28	C2 豬肉蛋漢堡	30
C3 鮪魚蛋漢堡	35	C4 培根蛋漢堡	25
D1 紅茶	15	D2 奶茶	20
D3 咖啡	30		

漢堡類(包含 C1，C2，C3，C4)搭配 D2 奶茶可以折價 5 元，B2 牛肉三明治搭配 D3 咖啡優惠價只要 49 元。

請幫瓊斯設計一支程式可以輸入一組客人點選的項目，各品項中間用一個空白鍵隔開，不會照順序，有折扣的東西不一定會排在一起。輸出為該組客人消費的總金額。

輸入

```
A2 D1 D2 D3 C1 B2 B1 B1 C4 C2
```

輸出

```
237
```

問題 6
字元編碼轉換
4 分

前言

Unicode 是電腦科學領域裡的一項業界標準。它對世界上大部分的文字系統進行了整理與編碼，使得電腦可以用更為簡單的方式來呈現和處理文字。在表示一個 Unicode 的字元時，通常會用「U+」然後緊接著一組十六進位的數字來表示這一個字元。

UTF-8 則是一種針對 Unicode 的可變長度字元編碼，它可以用來表示 Unicode 標準中的任何字元，且其編碼中的第一個位元組仍與 ASCII 相容，這使得原來處理 ASCII 字元的軟體無需或只需做少部份修改，即可繼續使用。UTF-8 編碼方式和範例如下：

Unicode 代碼範圍		UTF-8 位元組數	位元組 1	位元組 2	位元組 3	位元組 4
U+0000	U+007F	1	**0**xxxxxxx			
U+0080	U+07FF	2	**110**xxxxx	**10**xxxxxx		
U+0800	U+FFFF	3	**1110**xxxx	**10**xxxxxx	**10**xxxxxx	
U+10000	U+1FFFFF	4	**11110**xxx	**10**xxxxxx	**10**xxxxxx	**10**xxxxxx

Unicode 代碼		Unicode 以二進位呈現	UTF-8 以二進位呈現	UTF-8 以十六進位呈現
$	U+0024	0100100	**0**0100100	24
¢	U+00A2	000 10100010	**110**00010 **10**100010	C2 A2
€	U+20AC	00100000 10101100	**1110**0010 **10**000010 **10**101100	E2 82 AC
𤭢	U+24B62	00010 01001011 01100010	**11110**000 **10**100100 **10**101101 **10**100010	F0 A4 AD A2

輸入

請實做一個 Unicode 轉換成 UTF-8 的轉換程式。輸入為一個 Unicode 代碼：

範例 1:
```
U+24B62
```

範例 2:
```
U+0024
```

輸出

輸出為該 Unicode 代碼的 UTF-8 以 4 個十六進位位元組(Byte)呈現(無需包含空格)：

範例 1:
```
F0A4ADA2
```

範例 2:
```
00000024
```

問題 7
防酒駕軟體
5 points

前言

因為台灣酒駕頻傳，造成許多人家庭破碎，為了避免酒醉上路造成無法挽回的後果，我們打算開發一款 App 去計算人體血液的酒精濃度。依據台灣道路交通安全管理規則第 114 條規定，飲用酒類或其他類似物後，其吐氣所含酒精濃度超過每公升 0.5 毫克或血液中酒精濃度超過 0.05% 者，不得駕車，否則就會被依酒後駕車加以處罰。我們已經知道血液約為人體體重的 8%，正常成人每小時可以代謝 10 毫升酒精，請依下列所提供資料計算，一個成年人需要多少時間後，血液中的酒精濃度會低於 0.05% (500 ppm)。請以幾小時幾分格式回答。假設酒類與血液的密度都是 1(公克/毫升)。

輸入

輸入包含酒類數量，以小數點分隔，前值單位為手，一手為六罐，後值為罐數，第二個正整數值為每罐容量，單位為毫升，第三個正整數值為酒精濃度，單位為百分比，第四個值為人體體重，單位為公斤，到小數點以下第二位。

```
2.3 350 5 70
```

輸出

程式必須印出兩個正整數值，第一個單位為小時，中間以冒號「：」分隔，第二正整數值為分鐘。

```
25:59
```

問題 8
數字系統
5 分

前言

在日常的生活中，通常我們都使用10進位的數字系統，然而現在我們有一個數字系統是以7為基礎的數字系統。

表示 0
% 表示 1
$ 表示 2
& 表示 3
\ 表示 4
~ 表示 5
! 表示 6
@ 表示 -1

根據上面的數字系統，我們可以算出以下例題在10進位數字系統裡的代表數字：

%&# 代表 1*(7^2)+3*7+0=49+21+0 = 70
@!\$~ 代表 (-1)*(7^4)+6*(7^3)+4*(7^2)+2*7+5 = -2401+2058+196+14+5= -128
!!~ 代表 6*(7^2)+6*7+5 = 294+42+5 = 341

你的輸入將會是以7為基礎系統的數字,你的程式會幫忙把這個特別的數字系統轉換成以10進位代表的數字

輸入0代表輸入的最後 一行(The end of line)

輸入

```
!$&
\~@
@@#$
0
```

輸出

```
311
230
-390
```

問題 9
數日子
6 分

前言

日常生活中有些特別的日子總是讓我們大家非常地期待它的到來，例如期待著生日、畢業、旅行、周年慶，或著有些日子會讓我們不時地去數著已經過了多久，例如家裡的寵物養了多久、和知心的朋友認識了多久、搬了多久的新家。現在要請你來設計一個數日子的程式，這個程式會有二組日期的輸入 (Start Date 和 End Date)，它的格式分別為"西元的年/月/日" ("xxxx/xx/xx"，x 為0到9的數字)，這個程式會算出從 Start Date 到 End Date 共有幾天。

請注意:
End Date 要大於 Start Date ，如果條件不符，請輸出"Invalid Input"。
如果 End Date 和 Start Date 同一天 ，算為1天。

判斷閏年的步驟: (閏年的定義是2月多了一天，2月29日。所以整年多了一天，共366天)

1) 西元年可以被4整除，跳至步驟2)。否則請跳至步驟5)
2) 西元年可以被100整除，跳至步驟3)。否則請跳至步驟4)
3) 西元年可以被400整除，跳至步驟4)。否則請跳至步驟5)
4) 是閏年 (有366天)
5) 不是閏年(有365天)

輸入

範例1:
```
2013/03/05
2012/02/01
```

範例2:
```
2011/03/05
2012/03/06
```

範例3:
```
2013/08/05
2013/09/01
```

輸出

範例1:
```
Total Days: Invalid Input
```

範例2:
```
Total Days: 368 days
```

範例3:
```
Total Days: 28 days
```

問題 10
火星文翻譯器
6 分

前言

火星文是時下年輕人用於網路或簡訊時所使用的簡語。

小明的媽媽發現她不太了解小明想表達的是什麼，尤其是每當她跟小明透過網路傳簡訊溝通時。您能幫她嗎？

注意：翻譯過之文字不再翻譯。

[TEXTSPEAK]與[\TEXTSPEAK]之間為火星文辭典的內容，[LOGIN_TITLE]與[\LOGIN_TITLE]之間為欲翻譯的文字。輸入包含登入標題[TEXTSPEAK], [\TEXTSPEAK], [LOGIN_TITLE], [\LOGIN_TITLE]。此程式幫忙轉成正常用的語言。

輸入

```
[TEXTSPEAK]
143: I love you
2DAY: Today
4EAE: For ever and ever
AND: Any day now
AFAIK: As far as I know
AFK: Away from keyboard
ATM: At the moment
B/C: Because
B4: Before
BF: Boyfriend
GF: Girlfriend
BFN: Bye for now
BOL: Be on later
BRB: Be right back
BTW: By the way
DM: Direct message
DWBH: Don’t worry, be happy
F2F: or FTF Face to face
FB: Facebook
FF: Follow Friday
FTL: For the loss
FTW: For the win
FWB: Friends with benefits
FWIW: For what it’s worth
FYEO: For your eyes only
FYI: For your information
GLHF: Good luck, have fun
GR8: Great
HAK: Hugs and kisses
HAND: Have a nice day
HT: Hat tip
HTH: Hope this helps
IANAL: I am not a lawyer
IDK: I don’t know
IIRC: If I remember correctly
IKR: I know, right?
ILU: I love you
IMHO: In my honest opinion
IMO: In my opinion
IRL: In real life
IU2U: It’s up to you
```

```
IYKWIM: If you know what I mean
J/K: Just kidding
J4F: Just for fun
JIC: Just in case
JSYK: Just so you know
K: Okay
LMBO: Laughing my butt off
LMK: Let me know
LOL: Laughing out loud
MM: Music Monday
MSM: Mainstream media
NAGI: Not a good idea
NM: Never mind
NMU: Not much, you?
NP: No problem
NSFW: Not safe for work
NSFL: Not safe for life
NTS: Note to self
OH: Overheard
OMG: Oh my God
ORLY: Oh, really?
PAW: Parents are watching
PLZ: Please
PPL: People
PTB: Please text back
QQ: Crying
RAK: Random act of kindness
RL: Real life
ROFL: Rolling on the floor laughing
RT: Retweet
RUOK: Are you okay?
SMH: Shaking my head
SRSLY: Seriously
SSDD: Same stuff, different day
SWAK: Sealed with a kiss
SWYP: So, what's your problem?
TIA: Thanks in advance
TIME: Tears in my eyes
TMB: Tweet me back
TMI: Too much information
TMRW: Tomorrow
TTYL: Talk to you later
TU: Thank you
VSF: Very sad face
WB: Welcome back
WTH: What the heck?
WTPA: Where the party at?
WYCM: Will you call me?
YGM: You've got mail
YMMV: Your mileage may vary
YW: You're welcome
ZOMG: Oh my god
[\TEXTSPEAK]
[LOGIN_TITLE]
TU for using textspeak translator, HAND !
[\LOGIN_TITLE]
```

輸出

```
Thank you for using textspeak translator, Have a nice day !
```

問題 11
單車快遞
6 分

前言

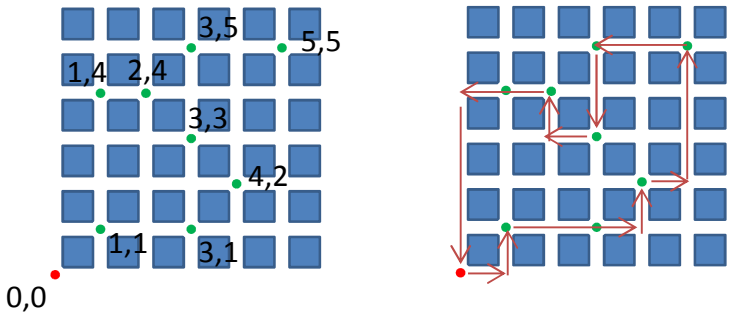
傑克是快遞員，每天的工作就是在 X 社區裡穿梭收送快遞。

X 社區是一個規劃方正的新住宅區，每個區塊距離都是相隔 100 公尺。

每天早上傑克都會收到需要收送的快遞地點清單，並且由快遞中心(0,0)出發，前往各個快遞點後回到快遞中心。

收送快遞的地點並沒有順序性，但是傑克希望能用最小的路徑完成他的工作。

以下圖為例，傑克前往了八個地點，總共騎了 2200 公尺的距離。



程式要求

請按照輸入的地點計算出傑克的最短路徑距離。每個地點以(x,y)表示，最後以(0,0)作為結尾.

輸入

```
(1,1)(1,4)(2,4)(3,1)(3,3)(3,5)(4,2)(5,5)(0,0)
```

輸出

單位：百米

```
22
```

問題 12
走迷宮
7 分

前言

請設計一個走迷宮程式給電腦鼠。

請注意：
起點以”-“代表，終點以”*” 代表。
請輸出電腦鼠行經的路徑所需的步數(起點與終點不列入計算)。

輸入

```
@@@@@@@@@@@@@@@@@@@@@
-                   @
@@@@@@@@@@@@@@@@@@@ @
@         *@       @ @
@@@ @ @@@@@ @@@@@@ @
@ @ @ @         @ @ @
@ @@@ @@@@@@@@@@ @ @
@     @     @    @ @
@ @@@@@ @ @ @@@@@@ @
@ @     @ @       @ @
@ @ @@@@@ @@@@@@@@ @
@ @    @    @       @
@ @@@@@@@@@@@@ @@@@@
@         @   @     @
@@@@ @@@ @@@@@@@@@ @
@  @ @            @ @
@ @  @@@@@@@@@@@@ @ @
@ @ @           @ @ @
@ @@@@@@@@@@@@@@   @
@                   @
@@@@@@@@@@@@@@@@@@@@@
```

輸出

```
Minimum distance: 82 steps
```

問題 13
解密碼
7 分

前言

馬克波特為了隱藏 HP 的商業機密，特地用了一種方式把英文字母給重新編碼。
編碼的規則如下：

1. 把字母從 A 到 Z 分別編號為 1 至 26 號 (A 為 1 號, B 為 2 號 … Z 為 26 號，以此類推)。
2. 每個單字為一組，假設開頭的第一個字母為 X 號，編碼後為 26-X+1 號。
3. 第二個字母開始，假設編號為 Y，前一個字母編碼之後的號碼為 Z。
    a. 若 Y > Z，則編碼後為：26 - (Y-Z) + 1
    b. 若 Y < Z，則編碼後為：(Z-Y)
    c. 若 Y = Z，則編碼不變

例如：HP 編碼為 SC，CODE WARS 編碼為 XIEE DCLT。

請依照解碼前相同對應的格式輸出，皆為大寫字母。

輸入

```
SC NMBXE GZU YTAH HCLQLUB RD GZU DPYMI
```

輸出

```
HP MAKES THE BEST SERVERS IN THE WORLD
```

問題 14
租屋電費計算
8 分

前言

有一房屋，房東將房屋分成兩個隔間(房間A 、房間B)分租給房客。 但只有一間有獨立電表(房間B)。另一個是整個房屋的總電表。台灣電力公司的電費計算只會有一張總電表的帳單。亦即

房間A 用電度數 = 總電表度數 － 獨立電表度數
房間B 用電度數 = 獨立電表度數

而台電電費計算方式採階梯式算法，用電量愈高的級距，會被採計愈高電價。下方為台電電費級距表

120度以下部分	每 度	2.10元
121~330度部分	每 度	3.02元
331~500度部分	每 度	4.39元
501~700度部分	每 度	4.97元
701度以上部分	每 度	5.63元

舉例1: 總電表用電500度，台電電費為:
120 * 2.10 + 210 * 3.02 + 170 *4.39 = 252 + 634.2 + 746.3 = 1632.5(小數點第一位， 四捨五入) = 1633

舉例2: 總電表用電753度，台電電費為:
120 * 2.10 + 210 * 3.02 + 170 *4.39 + 200 *4.97 + 53 *5.63 = 252 + 634.2 + 746.3 + 994 + 298.39 = 2924.89(小數點第一位， 四捨五入) = 2925

請寫一程式，公平地計算房間A與房間B的電費。以下輸入輸出範例為「**公平**」計算的方式所計算出的結果。程式需要讀取一固定格式與檔名檔案(**power.txt**)，用來讀取台電電費級距表，第一個值表示級距範圍(正整數且偶數)， 第二個值表示每度單價(固定小數點以下二位)，第一個值與第二個值中間有一空白鍵，0 0 表示結束。以下為電費級距表檔案內容範例：

```
0 2.10
120 3.02
330 4.39
500 4.97
700 5.63
0 0
```

輸入

輸入包括兩個整數。第一個值代表總電表度數，第二個值代表獨立電表度數(房間B)。
實例 1: 500 230
實例 2: 753 500
實例 3: 500 0

輸出

程式必須印出房間A電費與房間B電費(小數點第一位四捨五入， 程式的輸出值可容許與期望值有±1的誤差)。
實例 1: 904 728
實例 2: 831 2094
實例 3: 1633 0

問題 15
熱帶草原貓
8 分

前言

熱帶草原貓是非洲藪貓(Serval)和一般家貓的混血,，依照血統純度分為下列等級：
F1 為 藪貓 和家貓的後代
F2 為　 F1 和家貓的後代
F3 為　 F2 和家貓的後代 …依此類推

而每一代的售價為 F1 的價格除以本身的世代數：
F2 售價為 F1 售價除以 2， F3 售價為 F1 售價除以 3 …依此類推。

凱文迪普在非洲捕獲一隻母藪貓，想要經營熱帶草原貓的販賣生意，題目設定的規則如下：

1. 純種藪貓禁止販賣，F1 的價格為一隻 30240 元。
2. 每一隻母貓最多生育一次，一胎包含一隻公貓兩隻母貓。
3. 公貓無繁殖能力所以一出生就要賣掉，母貓可選擇賣掉或是生育，但同一世代所有母貓只有一種選擇。
4. 母貓生育之後就沒有價值無法販賣。繁殖需花費固定成本。

請問凱文迪普要讓貓繁殖到第幾代才能獲得最大利潤？總共賺多少？

輸入

輸入為一整數代表每隻母貓繁殖需花費的金額，範例如下：

```
8000
```

輸出

此程式必須印出能獲得最大利潤的繁殖世代數以及利潤金額。
以範例輸入為例，輸出如下：

```
F6 237888
```

問題 16
生命遊戲
9 分

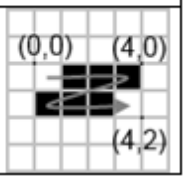
前言

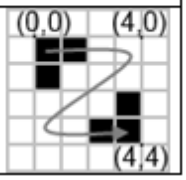
生命遊戲是個二維的細胞模擬程式，把一個無限的平面分割成很多方格，每一格子代表一個細胞，每一個細胞有八個鄰居。這些細胞有兩種狀態：生或死。活的細胞在方格內標記成黑色，而死的細胞則不塗色。這遊戲有世代演化的概念，某細胞在下個世代的狀態會與當前世代的週邊細胞狀態有關，規則如下：

- 若為活細胞
  - 當八個鄰近細胞中，只有零個或一個是活細胞，則該細胞會在下個世代因孤獨而死亡。
  - 當八個鄰近細胞中，恰有二或三個是活細胞，則該細胞在下個世代會繼續存活。
  - 當八個鄰近細胞中，有四個或超過四個是活細胞，則該細胞會在下個世代因擁擠而死亡。
- 若為死細胞
  - 當八個鄰近細胞中，恰有三個是活細胞時，則該細胞會在下個世代復活。

輸入

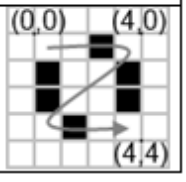
請實做生命遊戲。輸入為一組描述二維棋盤和所需要模擬的世代數的字串，以空白當區隔，第一個字串為**寬度**，第二個字串為**高度**，第三個字串為**細胞狀態**（1 代表生，0 代表死，由左上角為原點，先從左排至右，再從上排至下），第四個字串為需要模擬的**世代數**。

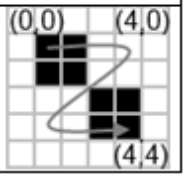
範例 1 輸入	示意圖
4 2 01111110 1	

範例 2 輸入	示意圖
4 4 1100100000010011 1	

輸出

輸出亦為一組描述二維棋盤的字串，格式同輸入的前三個字串。請注意輸出的長寬必須與**最小**可以描述結果的矩形相同。以上面的輸入為例，經過一個世代的演化，分別成為：

範例 1 輸出	示意圖
4 4 0010100110010100	

範例 2 輸出	示意圖
4 4 1100110000110011	

問題 17
行車時間
10 分

前言

因為智慧型手機和行動上網的普及，連帶可以利用手機進行車用導航。在行進中可以上網，下載地圖資訊和路況。有鑒於科技的進步，讓我們可以快速獲得即時的交通訊息，得知路況及雍塞路段，讓駕駛者可以避開繁忙的交通路段，節省行車時間。現在請各位研發一個程式，幫助駕駛者，判斷最快的行車路徑，讓駕駛可以在最短時間到達目的地。

不同的路段有著不同的限速，和不同的路況(這裡我們假定全天的路況是一樣的)。不同的路況會對能夠駕駛的實際速度造成影響，以下是影響程度:

Light: 可行駛至最高速限

Medium: 僅可行駛速限的 60%

Heavy: 僅可行駛速限的 30%

輸入

每個地點都是以一個字母代表，輸入的第一行是出發地點代號，第二行則是目的地代號。從第三行列出各地點之間的道路距離、限速，以及路況。 A B 30 50 Light 表示由 A 至 B 的道路是 30 km, 限速是 50 km/h, 而路況是 Light。由於地點以單一字母表示，所以限制在 26 個地點之內。程式輸入允許兩個相鄰地點之間連接的道路可以超過一條，這就好像台北到宜蘭可以走雪山隧道或北宜公路，但不會有從 A 點到 A 點的路線。輸入 0 代表結束。

```
A
F
A B 30 50 Light
A C 15 80 Heavy
A D 40 100 Medium
A B 30 60 Heavy
B C 20 60 Light
B E 60 80 Medium
C F 70 100 Light
D B 30 80 Medium
E F 15 40 Heavy
0
```

輸出

對每筆輸入，你的程式需要輸出一行資訊，列出最快且唯一的行車路線、中間經過的每一個點，以及總共花費的時間(分鐘數以下四捨五入)。輸出格式為 **小時:分鐘**，分鐘數小於 10 時請補 0，例如 2:05，而非 2:5。以範例輸入為例，輸出：

```
A C F 1:20
```

問題 18

無線環境下的資料散佈

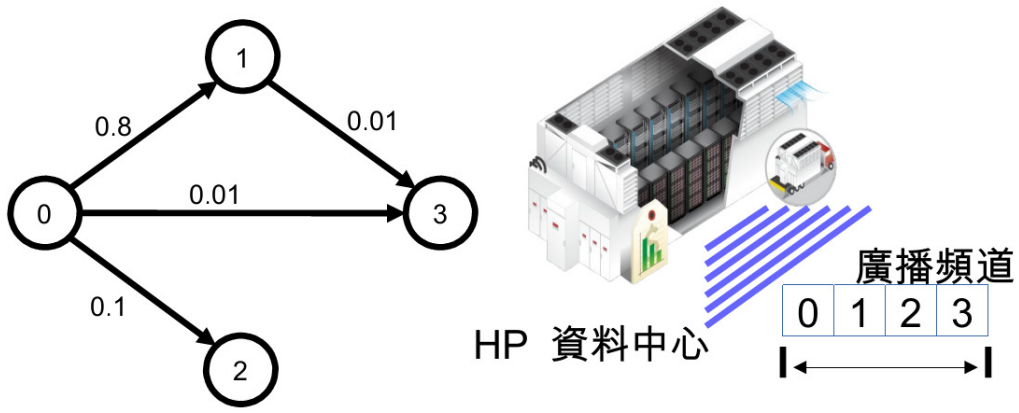
14 分

前言

無線環境下，透過無線頻道散佈資訊是一個有效率的方式。此種散播資訊的方式可以被廣泛應用，例如尋找最佳路徑的服務，查詢股票交易價格服務，或者查詢大眾資訊網頁服務等等。然而在這些服務中，資料彼此之間存在一些相依性。而這些相依性可以被量化並用機率來表示。假設 HP 的資料中心有 4 筆資料如下圖，且其編號分別是 0,1,2,3。箭頭表示用戶會存取的順序與路徑。所以這四筆資料可能被存取路徑可能如下：

- 路徑 A：先得到編號 0 的資料，再取得編號 1 的資料，然後停止收聽。
- 路徑 B：先得到編號 0，再取得編號 1，然後繼續收聽而得到編號 3 的資料，然後停止。
- 路徑 C：先得到編號 0 的資料，再取得編號 3 的資料，然後停止收聽。
- 路徑 D：先得到編號 0 的資料，再取得編號 2 的資料，然後停止收聽。
- 路徑 E：僅需要編號 0 的資料。

兩筆資料間若沒有箭頭存在，表示這兩筆資料沒有關聯也就是說使用者不可能會這樣下載資料。



HP 資料中心利用無線頻道的方式散播這四筆資料讓這些行動用戶去接收他們所需的資訊，然而雖然 HP 知道每筆資料之間的關係順序，但是需要這些資料的行動用戶很多。並且每個用戶需要得到這些資料的路徑也不盡相同，為了儘量讓每個用戶花比較少的時間得到他們想要的資料，HP 資料中心分析這些存取的路徑，並給予相對應的權重（機率）以方便做資料的排程。如上圖：

下載資料編號 0 後要繼續下載資料編號 1 的機率為 0.8。同理，下載資料編號 1 後要繼續下載資料編號 3 的機率為 0.01 以此類推。

HP 資料中心有了以上資訊後 （資料存取順序以及資料被存取的機率），如何安排資料廣播的排程就靠你來解決了。你必須找出一個排程並滿足下面條件：

1. 花費最少的時間將每一筆資料廣播出去
2. 廣播資料的順序必須滿足輸入給定的順序關係
3. 所有用戶平均等待時間要最少

定義與假設：

a. 資料彼此之間的關係不會存在迴圈。
b. 假設編號 0 是所有資料的源頭，所以被存取的機率是 1。
c. 假設無線頻道每一個時間單位裡面只能存放一筆資料。所以四筆資料需要四個時間單位才能在無線頻道中散佈出去。
d. 用戶存取特定資料的等待時間 ＝ 資料被存取的機率乘以該資料在排程中的位置
e. 所有用戶平均等待時間 ＝ 每筆資料被存取的等待時間之總和

以上面例子來說，HP 資料中心廣播這四筆資料的順序為 0,1,2,3。由此可知下載編號 0 資料需要 1 個時間單位，編號 1 需要 2 個時間單位，編號 2 需要 3 個時間單位，編號 3 需要 4 個時間單位。然而編號 0 被存取的機率為 1，編號 1 被存取的機率為 1*0.8 = 0.8，編號 2 被存取的機率為 1*0.1 = 0.1，編號 3 被存取的機率為 1*0.8*0.01 + 1*0.01 = 0.018（因為有兩個路徑可以取得編號 3）。所以編號 0 被存取的用戶等待時間 ＝ 1*1 = 1，編號 1 被存取的用戶等待時間為 0.8*2 = 1.6，編號 2 被存取的用戶等待時間為 0.1*3 = 0.3，編號 3 被存取的用戶等待時間 0.018*4 = 0.072。因此所有用戶平均等待時間 ＝ 1 + 1.6 + 0.3 ＋0.072 = 2.972。

為了方便用（x, p, y）來表示兩筆資料的關係：取得 x 資料後繼續取得 y 資料的機率為 p，(0,0,0) 為結束。請輸出滿足題目中條件的排程順序(只會提供有唯一解的輸入)。

輸入

```
(0,0.8,1) (1,0.01,3) (0,0.01,3) (0,0.1,2) (0,0,0)
```

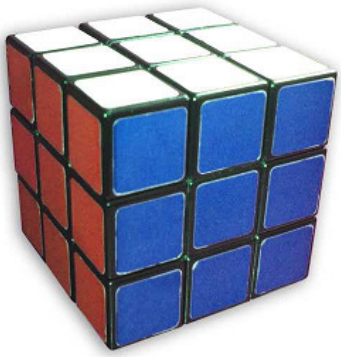
輸出

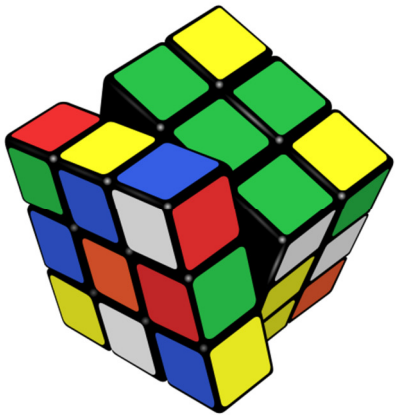
```
0,1,2,3
```

問題 19
魔術方塊
15 分

前言

魔術方塊，在中國大陸稱為魔方，在香港稱為扭計骰，為由匈牙利建築學教授暨雕塑家魯比克·厄爾諾於 1974 年發明的機械益智玩具，最初的名稱叫 Magic Cube，1980 年 Ideal Toys 公司於販售此玩具，並將名稱改為 Rubik's Cube。





輸出的表現方式

本題魔術方塊採用官方配色, 初始狀況如下圖:

前面(F): 綠色 G

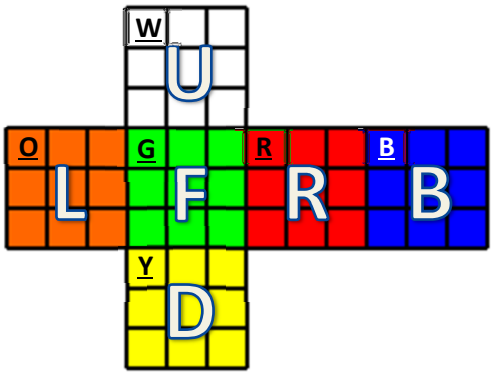
後面(B): 藍色 B

左面(L): 橘色 O

右面(R): 紅色 R

上面(U): 白色 W

下面(D): 黃色 Y



標準輸出表示法為以九宮格為一組, 從左到右顯示前, 後, 左, 右, 上, 下各面九格之顏色, 各組間以一個空白隔開。

初始狀態的表示法如下:

```
GGG BBB OOO RRR WWW YYY
GGG BBB OOO RRR WWW YYY
GGG BBB OOO RRR WWW YYY
```

輸入的表現方式

魔術方塊的轉動過程(輸入)，會用 Singmaster 符號來書寫（由 David Singmaster 發明）。

書寫方式如下：

- 若是順時針旋轉，則直接寫上符號；若是逆時針旋轉，則在符號後加上「i」；若是旋轉 180°，則在符號後加上「2」。
- F、B、L、R、U、D 分別代表前、後、左、右、上、下層。
- f、b、l、r、u、d 分別代表前、後、左、右、上、下兩層，代表連中間層一起轉。
- x、y、z 分別代表將整個魔術方塊做 R、U、F，因為在速解魔術方塊的時候，並不會總是將一個面朝向自己。
- M、E、S 代表旋轉中間層，分別相當於 lLi、dDi、fFi。

程式要求

請按照輸入的魔術方塊轉動過程，計算出各面從初始狀態轉動後之結果。

輸入

(1) `FRB`

(2) `Urz`

(3) `FLrU2xdiMluDzE`

輸出

(1)

```
GGR WWO GOY WWB WRR RRB
GGY BBB WOY RRY WWG YYB
GGY BBB WOY RRY OOG OOO
```

(2)

```
GGR OBB YYY WWW OOG RRR
YYY WWW OBB GGR OOG RRR
YYY WWW OBB GGR OOG BBB
```

(3)

```
BBR WOG RYR YYG WWO OWB
RYR YWY GBG BGR BOO GRR
YGO GBB WOG YWY WOB OWR
```

問題 20
俄羅斯方塊
16 分

前言

俄羅斯方塊(Tetris)在 1984 年由前蘇聯人 Alexey Pajitnov 所發明。自從問世以來，俄羅斯方塊一直受到大家的喜愛。從俄羅斯方塊的設計上衍生出許多變化形的遊戲，玩家間也一直有各種速度和分數的挑戰。可以算得上是益智遊戲的代表之一。

俄羅斯方塊的基本方塊一共有以下七種(圖片取自 Wikipedia)：

基本的遊戲方式是在寬度 10 格，高度不一定的空間內。每次會有一個方塊出現，而玩家需要將它排列以填滿完整的橫列才能消去。考慮到時間的複雜度，我們限制空間為寬度 6 格，高度 6 格的正方形。不需要考慮移動的情況，只需要計算方塊任意旋轉後放進區域內的情況。

輸入

你的程式需要處理來自標準輸入(STDIN)的資料。每筆輸入會在一列之中包含一串方塊的代號，請求出這串方塊可以消去幾行。方塊不會超過 6 個。輸入範例如下：

```
範例 1:
STZI
範例 2:
OOO
範例 3:
JLI
範例 4:
II
```

輸出

對每筆輸入請輸出一個數字代表總共可以消去的列數。不需要考慮方塊出現的順序或是移動時會不會卡住的情況。如果無法消去，請輸出 0。輸出範例如下：

```
範例 1:
2
範例 2:
2
範例 3:
2
範例 4:
0
```

問題 21
數獨
17 分

前言

這幾年來，我們可以常常在報紙上看到數獨(Sudoku)的小遊戲，數獨也流行地成為打發時間的遊戲，不管男女老少大家都可以玩。

你將會設計一個標準的3x3數獨解題程式。

在標準的3x3數獨裡，遊戲的規則是把1到9的數字填在9x9的方格裡。
遊戲規則:

1) 在9x9方格中再分成3x3的9個大方格。每個大方格裡又有9個小方格，小方格裡可以放1~9的某個數字，但不能重複。
2) 每一列有9個小方格，放1~9的某個數字，但不能重複。
3) 每一行有9個小方格，放1~9的某個數字，但不能重複。
4) 不用考慮因為空格數太多，造成有無限多解情況，測試資料會確保只有唯一解。

輸入

```
---26-7-1
68--7--9-
19---45--
82-1---4-
--46-29--
-5---3-28
--93---74
-4--5--36
7-3-18---
```

輸出

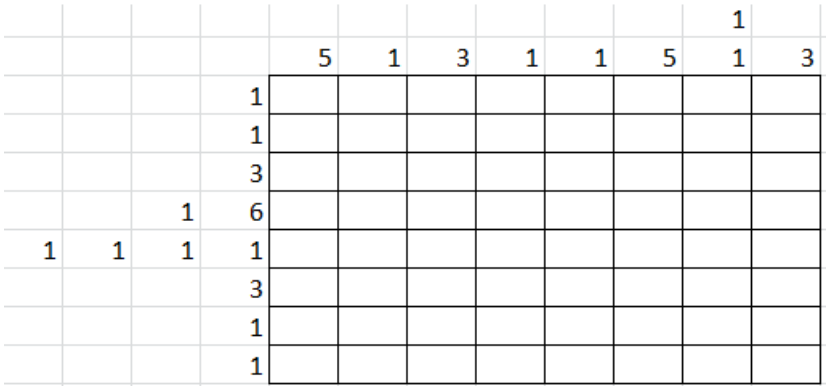
```
435269781
682571493
197834562
826195347
374682915
951743628
519326874
248957136
763418259
```

問題 22
邏輯馬賽克
19 分

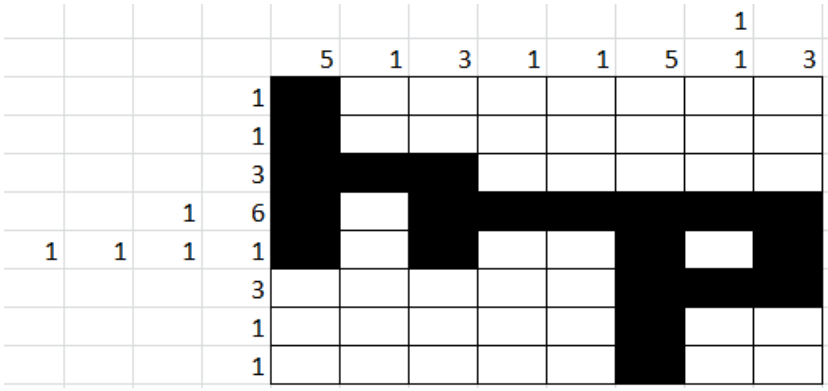
前言

邏輯馬賽克(Nonogram) 是一種由日本人發明的邏輯解謎遊戲，是一種根據數字提示填滿方格的謎題。它又被稱為 picross，並有同名的電玩遊戲。每行或列的數字代表有幾格連續填滿的方格，多組數字則表示不連續的填滿。例如 2 1 代表本行(或列)中有兩組填滿的方格，第一組是兩格，第二組是一格。兩組中間間隔可能是一格或更多。以下是範例問題：

問題：



解答：



這個題目希望各位以程式的方式來解邏輯馬賽克的問題(裁判會使用只有唯一解的輸入進行驗證)。

你的程式必須要在一定時間內解出答案。需要解答的圖形會限制在 8*8 以內的矩形。輸入的第一行是行數(x)與列數(y)，接著以左上角開始，先橫後直的方式輸入每一行的數字，以空格隔開，完全沒有需要填滿的行或列值則是 0。以範例問題為例，輸入如下：

輸入

```
8 8
5
```

```
1
3
1
1
5
1 1
3
1
1
3
1 6
1 1 1 1
3
1
1
```

輸出

輸出格式請以「@」代表填滿的格子，「.」代表空白的格子。範例輸出如下：

```
@.......
@.......
@@@.....
@.@@@@@@
@.@..@.@
.....@@@
.....@..
.....@..
```