0_送分題 – Hello World

(30分)

前言

比賽開始了！

趕快驗證一下，

網路是否設定正確?

上傳競賽程式是否順利?

程式解答是否用 STDOUT 輸出？

都沒問題，30分就到手了！ 繼續 ... 衝！衝！衝！

問題敘述

請寫一個程式輸出Hello World!

輸入格式

本題無需輸入值

輸出格式

[A~Z][a~z], 空格, 以及及常用英文符號 。

資料範圍

[A~Z][a~z], 空格, 以及驚嘆號 “!”

輸入範例1

(無輸入值)

輸出範例1

Hello World!

範例解釋

輸入範例1, 無輸入值，簡單而快樂的輸出Hello World!

1_佈告欄貼紙 (Stickers on Bulletin)

(5分)

問題敘述

小明的班級在新的學期搬到了一間新的教室。班上的同學採購了一堆貼紙，想用來佈置教室後面的佈告欄。身為學藝股長的小明，想到可以把「他們班的班號」張貼紙，排成一個長方形，這樣貼到佈告欄上後不但比較整齊與壯觀，而且還同時具有他們班的特色。

然而，佈告欄終究會有長度與高度上的限制，所以如果長方形的長超過了佈告欄的長度，或是寬超過了佈告欄的高度的話，那這個長方形就無法被貼在佈告欄上。在這裡，每張貼紙的長和寬都是 1。

由於給定貼紙數量（也就是他們的班號）之後，能夠排成的長方形的長寬組合可能還是有很多種，加上要考慮到佈告欄的大小，因此小明希望你能幫他寫個程式，來列出所有可行的長方形。

輸入格式

輸入只有一行，包含三個整數：N、L、H，三者會用空白鍵分開。

其中，N 代表小明班的班號，L 和 H 分別代表佈告欄的長度與寬度。

輸出格式

輸出要有 x 行，x 是指可行的長方形種類的數量。

在輸出的第 i 行中 ($1 \leq i \leq x$)，要印出兩個整數：l_i 和 w_i，兩者用空白鍵分隔。其中，l_i 和 w_i 分別表示其中一種可行的長方形的長和寬。此外，你的輸出必須滿足 $l_1 < l_2 < ... < l_x$。

資料範圍

$1 \leq N \leq 1000.$
$1 \leq L \leq 1000.$
$1 \leq H \leq 1000.$

輸入範例 1

```
104 20 20
```

輸出範例 1

```
8 13
13 8
```

輸入範例 2

```
220 50 15
```

輸出範例 2

```
20 11
22 10
44 5
```

輸入範例 3

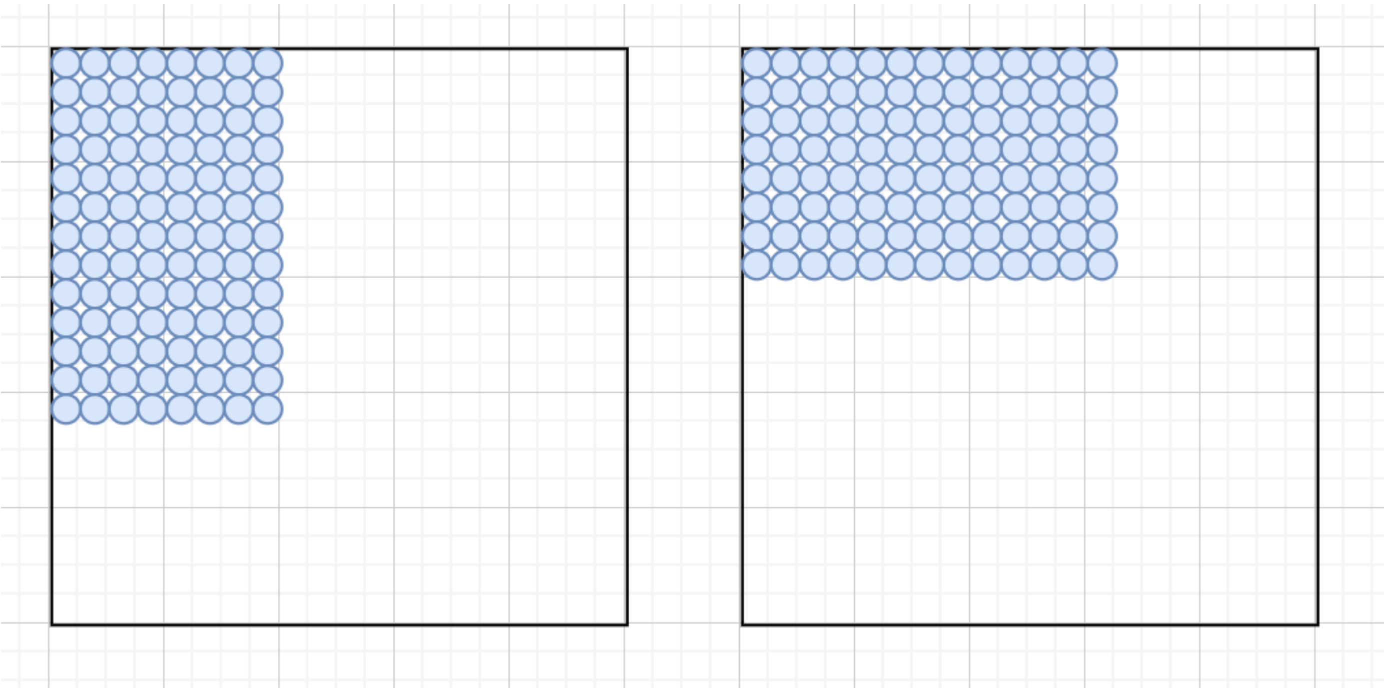
```
327 10 1000
```

輸出範例 3

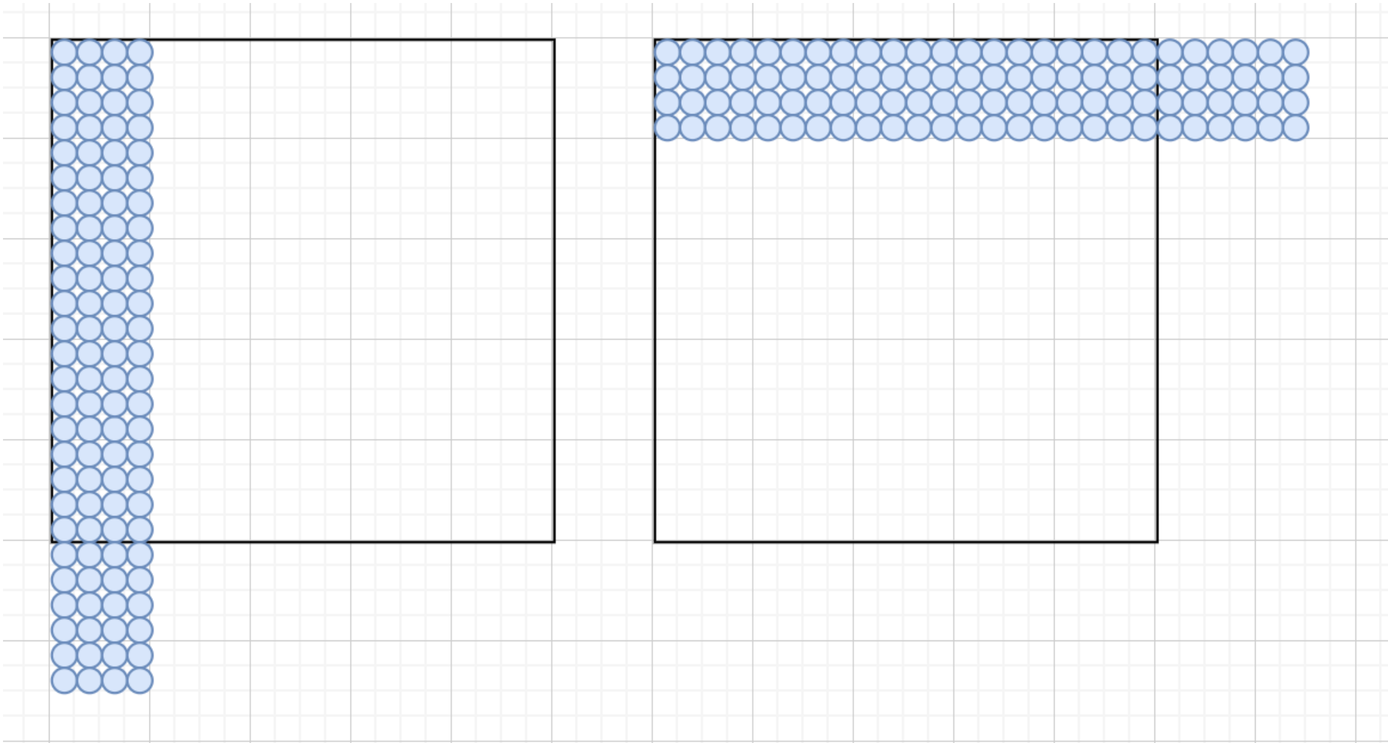
```
1 327
3 109
```

範例說明

在範例一中，班號是104，因此長方形的長與寬可以是(1,104)，(2,52)，(4,26)，(8,13)，(13,8)，(26,4)，(52,2)，(104,1)。但是因為佈告欄的長度與高度都只有20，所以只有(8,13)和(13,8)兩者是可行的，如下圖。



而其他的長方形都是不可行的，因為會超出邊界。（下圖是(4,26)和(26,4)的長方形）



在範例二中，班號是220，因此長方形的長與寬可以是(1,220)，(2,110)，(4,55)，(5,44)，(10,22)，(11,20)，(20,11)，(22,10)，(44,5)，(55,4)，(110,2)，(220,1)。但是因為佈告欄的長度只有50，高度只有15，所以只有(20,11)和(22,10)和(44,5)三者是可行的。

在範例三中，班號是327，因此長方形的長與寬可以是(1,327)，(3,109)，(109,3)，(327,1)。但是因為佈告欄的長度只有10，高度只有1000，所以只有(1,327)和(3,109)兩者是可行的。

2_捉出桌初升降賽 (Table Tennis)

(10分)

問題敘述

桌球王牌鄭教授完全不會打桌球、連桌球拍都不會拿的鄭笨蛋都是YTP學校的學生。

這學期，他們選到了同一堂桌初(桌球初級)的課程，第一堂課老師為了找出桌初班級內的真強者，舉辦了一場捉出桌初升降賽，規則如下：

桌球場內共有 N 張桌子編號 1 到 N ，一開始班上的 $2N$ 個人會被隨機分配到一張桌球桌旁邊，每張桌球桌旁邊各兩人。在每一輪比賽中，被分配到同桌的人會對打一場桌球，決定贏家或輸家。一輪比賽結束後，在第 i 桌比賽的贏家會往前移動一桌到第 $i-1$ 桌，而第 i 桌比賽的輸家則會往後移動一桌到第 $i+1$ 桌。比較特別的是在第一桌的贏家無法往前移動，故只需停留在原地。最後一桌的輸家同理也只需停留在原地。不難驗證在移動結束後每桌旁邊都會有恰好兩個人，這時開始下一輪比賽。

一開始在第 t_1 桌的鄭教授想找一開始在第 t_2 桌的鄭笨蛋比賽，但因為距離遙遠且上課時間有限，鄭教授想知道最少需要幾輪比賽才能讓他跟鄭笨蛋同桌競技。他可以遠距遙控鄭笨蛋讓他在某一輪輸球或贏球，也可以自己裝弱決定他這場要輸球或贏球。請幫鄭教授算算看最少要幾輪比賽才能讓他們相遇！

輸入格式

輸入資料只有一行，共有三個正整數 N, t_1, t_2 ，如題目敘述所示， N 代表桌球場內共有幾張桌球桌， t_1, t_2 分別代表鄭教授與鄭笨蛋一開始所在的桌球桌編號。

輸出格式

輸出一個整數於一行，代表兩人最少需要幾輪比賽才能相遇。

資料範圍

$1 \leq N \leq 10^9$

$1 \leq t_1, t_2 \leq N$

範例輸入1

```
5 3 1
```

範例輸出1

```
1
```

範例輸入2

```
6 3 6
```

範例輸出2

```
2
```

範例輸入3

```
4 2 2
```

範例輸出3

```
0
```

範例說明

Case 1. 若鄭教授在第一輪贏球且鄭笨蛋在第一輪輸球，他們會同時出現在第2桌。
Case 2. 若鄭教授在第一輪輸球且鄭笨蛋在第一輪輸球，鄭教授在第二輪輸球且鄭笨蛋在第二輪贏球。他們會同時出現在第5桌。
Case 3. 一開始他們就在同一桌。

3_純友誼 (Friendship)

(10分)

問題敘述

小得是小里很要好的朋友，再過幾天就要到小得的生日了，身為數學迷小得的摯友，小里決定準備一組對戒，並將一組「友好數（amicable numbers）」分別刻在兩枚戒指上，一個送給小得，一個自己留著，象徵兩人堅貞不渝的友誼。

甚麼是友好數呢？友好數指的是兩個正整數中，彼此的全部正因數之和（本身除外）與另一方相等。舉例來說，220 的所有正因數（除了自己）是 $1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110$ 把這些數字加總我們會得到 284，至於 284 的所有正因數（除了自己）是 $1, 2, 4, 71, 142$，加總後會得到 220，所以 220 和 284 是一組友好數。

然而小里想要找找看有沒有其他的友好數，並挑出一組最喜歡的刻在戒指上，但他並不知道該如何尋找友好數，所以他請身為程式大師的你幫他寫一支程式來協助找出小里所提出的數字的友好數。

輸入格式

輸入只有一個正整數 N，如題目敘述所示，代表小里所提出的數字。

輸出格式

輸出共一行，如果 N 有友好數，則輸出 N 的友好數，如果 N 沒有友好數，則請輸出 "Not found" （不含雙引號）。

資料範圍

- $1 \leq N \leq 10^{12}$

輸入範例1

```
220
```

輸出範例1

```
284
```

輸入範例2

```
6
```

輸出範例2

```
Not found
```

輸入範例3

```
12
```

輸出範例3

```
Not found
```

範例說明

- 在範例一中，220 的所有正因數（除了自己）是 $1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110$，而且 $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$，而 284 的所有正因數（除了自己）是 $1, 2, 4, 71, 142$ 加總得到 $1 + 2 + 4 + 71 + 142 = 220$ 可知 284 是 220 的友好數，所以輸出 284。
- 範例二中，6 的所有正因數（除了自己）是 $1, 2, 3$，總和為 6，然而自己不能當自己的朋友，所以沒有任何數字是 6 的友好數。
- 範例三中，12 的所有正因數（除了自己）是 $1, 2, 3, 4, 6$，總和為 16，至於 16 的所有正因數（除了自己）是 $1, 2, 4, 8$，總和為 15，所以 12 並沒有友好數。

4_高度危機 (Multi-level subtractions)

(10分)

問題敘述

首先，簡單複習國中數學！

$x^n = x$ 連乘 n 次,

等差數列的第 n 項的就是 $x_n = x_1 + (n-1) \times d$，其中 d 是公差，n 是第幾項，

等比數列的第 n 項就是 $a_n = a_1 \times r^{(n-1)}$。

在這一題中，會給定你一個等差數列初值 x_1 ，公差 d 及次方 p，從而可以透過這三個數得到一個等差數列的次方數列，也就是$x_1^p, (x_1+d)^p, \ldots, (x_1+(n-1)\times d)^p$。

舉個實際的例子吧！假設 $x_1 = 1, d = 2, p = 2$ ， 可以先透過 $x_1 = 1, d = 2$ 得出一個等差數列 $1, 3, 5, 7, \ldots$ ， 接著再對每個數字求 $p = 2$ 次方，會變成 $1, 9, 25, 49, \ldots$ 。

得到這個序列後，再將**相鄰的數字相減（大減小）**，直到所有數字相同為止。以上述例子來說，對 $1, 9, 25, 49, \ldots$ 做一次**相鄰的數字相減**後會得到 $8, 16, 24, \ldots$ ，若再做一次則可以得到 $8, 8, 8, \ldots$，此時因為所有數字皆相同，所以便不再進行**相鄰的數字相減**操作了！

已知任何公差及次方的等差數列，經過多次**相鄰的數字相減**運算一定會得到一個固定值，由於小明花了很多時間在計算而產生了高度危機，所以請幫忙寫一支程式在輸入不同的等差數列初值 x_1 ，公差 d 及次方 p，並輸出一個整數，代表做完數次**相鄰的數字相減**後的固定值。

輸入格式

輸入包含三個正整數於一行， x_1, d, p 以逗號分開，分別代表等差數列初值、公差及次方。

輸出格式

輸出一個整數於一行，代表做完數次差分後的固定值。

資料範圍

- $1 \leq$ x_1, d, p ≤ 6
- 題目保證，做完差分後的固定值是一個介於 1 和 $2^{31} - 1 = 2147483647$ 間的整數。

輸入範例1

```
1,2,2
```

輸出範例1

```
8
```

範例說明1

同題目敘述例舉說明！

輸入範例2

```
1,2,3
```

輸出範例2

```
48
```

範例說明2

第一步可以先得到等差數列 $1, 3, 5, 7, 9, \dots$。

接著取 3 次方 $1^3,\ 3^3, 5^3, 7^3, 9^3, \dots$ ，可得一數列: $1, 27, 125, 343, 729, \dots$ 。

經過第一輪的鄰近數值相減可得: $26, 98, 218, 386, \dots$。

經過第二輪的數值相減可得: $72, 120, 168, \dots$。

經過第三輪的數值相減可得 $48, 48, \dots$，故輸出 48。

輸入範例3

```
6,6,6
```

輸出範例3

```
33592320
```

5_選特別號 (Price of chosen numbers)

(15分)

問題敘述

家裡的電話、手機的號碼和車輛的車牌都是日常生活中有見到的數字，在申請時可以依喜好選特定的號碼，也因此特別或連續的號碼也就有一個基礎的額外金額，有些甚至會拍賣競價，並在這類系統中會有一隻程式負責設定這些號碼的基本價格。

為了簡化計算的時間, 我們設定金額分為 $1, 2, 5, 10, 20, 50$ 等單位, 規則、號碼範例及單位金額如下:

規則	範例	單位
非特殊號	135246, 505152	1
連續三個號碼相同或順號	555345, 237659	2
連續兩組相同的三個號碼	111222, 800800 333777, 955955	5
連續四個號碼相同或順號	123400, 123333	10
連續五個號碼相同或順號	644444, 900000	20
連續六個號碼相同或順號	456789, 888888	50

註: 順號是指數字依 0 到 9 上升, 或 9 到 0 下降的連串數字, 例如: 012 和 987 都屬於順號, 而 901 和 468 都不是順號。

試寫一程式讀取多筆 6 位數號碼, 然後計算出所有號碼的金額總和。

輸入格式

- 一個(utf-8) 格式的文字檔, 其中每一行有一個 6 位數號碼。
- 結尾行，非6位數號碼

輸出格式

輸出一個整數於一行，代表總單位和，值小於 50001 。

資料範圍

不含尾端換行字元有 6 位數字，若不是6 位數字則印出當前結果，並直接結束程式。

資料筆數小於等於 1000 筆。

輸入範例1

```
888888
135246
147258
555345
end
```

輸出範例1

```
54
```

輸入範例2

```
237659
111222
800800
123400
123333
644444
end
```

輸出範例2

```
52
```

輸入範例3

```
900000
456789
909090
901357
000000
end
```

輸出範例3

```
122
```

範例說明

- 範例1：50+1+1+2=54

6_心東徉要賣太陽眼鏡?! (SING TONG YANG)

(15分)

問題敘述

由於近日夏天炎熱，肉乾食品公司心東徉，決定開拓市場，插旗太陽眼鏡行業。但是礙於對新市場不熟悉，老闆阿東跟老闆娘小徉開始討論該如何行銷。
聰明的阿東想到可以利用現在心東徉熱賣的閃光肉包來配合宣傳太陽眼鏡，於是他決定跟小徉舉辦一場大胃王活動。在活動中，每位參賽者會有D分鐘的時間去吃閃光肉包，但每個肉包都必須在一分鐘以內吃完。參賽者可以選擇這一分鐘是要吃，還是要休息。最後D分鐘結束後，只要參賽者有連續吃至少三個閃光肉包的話，就會贏得一副太陽眼鏡。
現在阿東和小徉希望你可以幫助他們，計算出在D分鐘內，參賽者總共會有多少種方式，贏得太陽眼鏡?

輸入格式

輸入一個數字 D ，代表比賽進行 D 分鐘。

輸出格式

輸出一個數字 Y MOD 10^9+7，其中 Y 代表總共有多少種方式可以獲得太陽眼鏡。

資料範圍

- $0 \le D \le 100000$

輸入範例 1

```
1
```

輸出範例 1

```
0
```

輸入範例 2

```
3
```

輸出範例 2

```
1
```

輸入範例 3

```
4
```

輸出範例 3

```
3
```

範例說明

範例一：無法連續吃到三個閃光肉包

範例二：(吃，吃，吃) => 一種

範例三：(吃 , 吃 , 吃 , 休息) , (休息 , 吃 , 吃 , 吃) ， (吃，吃，吃，吃) => 三種

7_快速報名 (Quick Waiting Line)

(10分 /10分)

問題敘述

由T大資工系學生主辦，歷史悠久、最專業的程式競賽營隊 YTPCamp 就要開始報名了！身為程式營隊，YTPCamp 的報名方式當然是...現場報名！T大資工系館—DT館的所有門，屆時便會作為報名窗口，由工作人員進行報名資料收取與審核。

每頁資料的審核需要耗時 1 分鐘，當人數一多起來，可能排隊就要排上幾小時。其中最令人沮喪的莫過於，明明只有 1 頁報名資料要審，但前面有一個人拿著 100 頁資料等著審核。

身為 YTPCamp 的總召，小 C 感到十分緊張，怕學生們等太久而生氣。因此他預先把DT館眾多門口的一部份設定為快速通關，讓報名資料頁數比較少的學員，可以利用快速通關門優先審核，減少總等待時間。

資工系館有 N 道門可以排隊，其中有 M 道門是僅限資料頁數不超過 K 的學員才能排的快速通關門，學員的總等待時間的定義為每位學員的等待時間之總和，而一位學員的等待時間，便是排在前面所有人的報名資料頁數總和。

在正式開始報名的前幾小時，迫不及待想要報名的學生們早已經陸陸續續的來到了DT館前。聰明的學員們，擁有能快速辨別最小等待時間隊伍的能力，因此總會在能選擇的門中，選擇所需排隊時間最少的一道門前排隊（若有兩道門所需排隊時間相等，則會優先選擇快速通關門）。

小 C 已經知道了依照時間順序來到系館前的每位學員的報名資料頁數 a_i，你能幫小 C 算算，這些學員的總等待時間會是多少，好讓他來得及去買飲料補償他們嗎？

輸入格式

輸入第一行有三個整數 N, M, K，代表總共有 N 個門，其中 M 個是快速通關門，且資料頁數不超過 K 的學員可以排在快速通關門前。

第二行有一個整數 A，代表門前總共有 A 名學生在等待。

第三行有 A 個整數，第 i 個整數 a_i 代表第 i 名抵達系館的學生有 a_i 頁資料。

輸出格式

請輸出一個整數，代表學生們的總等待時間。

資料範圍

- $2 \leq N \leq 10000$
- $1 \leq M < N$
- $1 \leq K \leq 1000$
- $1 \leq A \leq 100000$

- $1 \le a_i \le 1000$

子任務

- 子任務 1 滿足 K=1000 (佔 10 分)。
- 子任務 2 沒有特別限制(佔 10 分)。

輸入範例 1

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

輸出範例 1

```
8
```

輸入範例 2

```
5 1 5
10
10 9 8 7 6 5 4 3 2 1
```

輸出範例 2

```
38
```

輸入範例 3

```
5 3 1000
5
4 4 3 1 2
```

輸出範例 3

```
0
```

範例說明

- 在範例 1 中，第 1 人排在了一般門前，等待時間為 0。

  第 2 人排在了第二扇的一般門前，等待時間為 0。

  第 3 人排在了快速通關門前，等待時間為 0。

  第 4 人排在了第 2 人的後面，等待時間為 3。

  第 5 人排在了第 1 人的後面，等待時間為 4。

  第 6 人排在了第 3 人的後面，等待時間為 1。

  因此，總等待時間為 8。

- 在範例 2 中，第 1, 2, 3, 4, 6 人前面沒有人，等待時間為 0。(第 6 人在快速通關門前)

  第 5 人排在了第 4 人的後面，等待時間為 7。

  第 7 人排在了第 5 人的後面，等待時間為 5。

  第 8 人排在了第 3 人的後面，等待時間為 8。

  第 9 人排在了第 7 人的後面(優先選擇快速通關門)，等待時間為 9。

  第 10 人排在了第 2 人的後面，等待時間為 9。

  因此，總等待時間為 38。

- 在範例 3 中，所有人前面都沒有人，總等待時間為 0。

8_最大和 (Maximum Sum)

(20 分)

時間限制: 1 second
記憶體限制: 512MB

問題敘述

在很多題目中，由於答案會很大，因此會需要先模一個數字 P 後再輸出你的答案。
而這題也不例外。
你有 N 個正整數 $A_1, A_2, \cdots, A_N$。
你可以選其中一些數字。令選到的數字的和為 X。
請找出 $X \mod (10^{17}+3)$ 最大可以到多少。

```
A={
  43257517791815812,
  7158485778091600,
  22932684354088977,
  26557122523572685,
  94189552430929
}

A1+A2+A3+A4=99905810447569074

(A1+A2+A3+A4+A5) mod 100000000000000003=0
```

輸入格式

輸入的第一行包含了一個整數 N ，代表有幾個正整數。
輸入的第二行包含了 N 個正整數 $A_1, A_2, \cdots, A_N$。

輸出格式

輸出包含一個正整數，代表 $X \mod (10^{17}+3)$ 最大可以到多少。

資料範圍

- $1 \leq N \leq 35$
- $1 \leq A_i \leq 10^{17}+3$

輸入範例 1

```
1
1
```

輸出範例 1

```
1
```

輸入範例 2

```
2
50000000000000002 50000000000000003
```

輸出範例 2

```
50000000000000003
```

輸入範例 3

```
5
43257517791815812 7158485778091600 22932684354088977 26557122523572685 94189552430929
```

輸出範例 3

```
99905810447569074
```

範例說明

- 在範例一中， X只有可能是 1 ，因此 $X \mod (10^{17}+3)$ 的最大值也是 1。
- 在範例二中， X 的可能值為 $(50000000000000002+50000000000000003)=10^{17}+5$，$50000000000000002$，$50000000000000003$，因此 $X \mod (10^{17}+3)$ 的最大值為 50000000000000003。
- 在範例三中，你應該選 A_1, A_2, A_3, A_4，它們的和為 99905810447569074。

9_登山王阿麥 (Mai The MTB Guy)

(20分)

時間限制：3秒

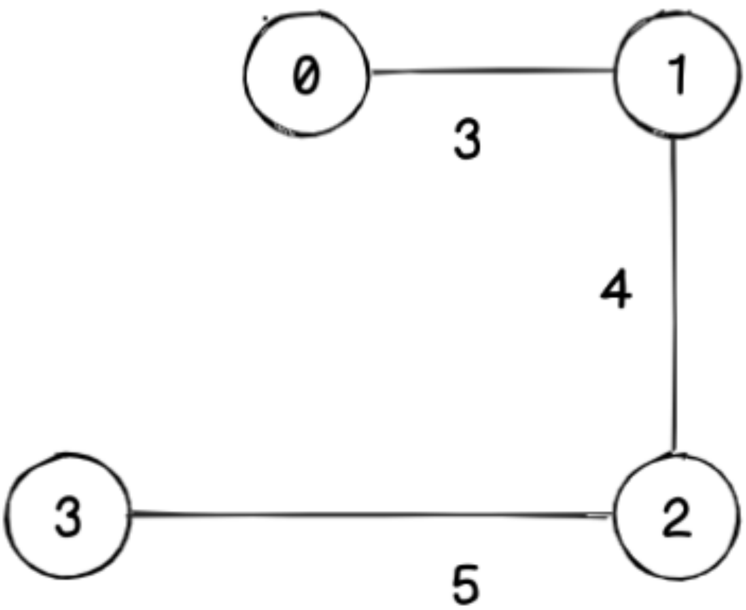
記憶體限制：512 MB

問題敘述

威廉麥每天騎著他的登山車到處遊蕩，因此他又別名「登山王阿麥」。作為瘋狂自行車俱樂部的一位資深成員，阿麥想要推廣他熱愛的登山車運動。推廣的方式為：找到這個國家最瘋狂的登山車路線，並且把這條路線完成。

在威廉麥的國家中，任兩座山峰都有一條路徑連接這兩座山峰，也就是說，對任兩座山峰X, Y，阿麥一定可以從X騎車騎到Y。此外，假設國家中總共有E條路，那麼國家中必定恰好有$E+1$座山峰。

如何決定登山車路線的瘋狂程度呢？阿麥一開始會先挑選這個國家的兩座山峰，然後計算這兩座山峰之間的距離。兩座山峰之間的距離是由他們之間的最短路徑決定，以下圖為例：



其中圓圈代表的是山峰，圓圈內的數字是山峰的編號，圓圈之間的線段代表連接兩座山峰的道路，旁邊的數字代表的是這條道路的長度。

在山峰 0 和 3 之間的最短路徑是 $(0 \to 1 \to 2 \to 3)$，因此這兩座山峰的距離為 12。可以簡單檢查一下在這兩座山峰之間沒有更短的路徑。這條路徑我們又把他叫做**瘋狂路徑**，因為他的長度恰好跟這兩座山峰的距離一樣。

傳說中，這個國家內最長的瘋狂路徑有一股神秘的魔法力量，而這條最長的瘋狂路徑的距離又稱作**魔法瘋狂距離**。

請幫登山王阿麥找到這個國家的魔法瘋狂距離，讓他可以專心準備這趟旅程。

輸入格式

第一行包含一個整數 E 分別代表這個國家的道路數。接下來的 E 行，每行包含 3 個整數 a_i, b_i, c_i 代表一條道路。整數 a_i, b_i 代表這條道路的兩個端點，整數 c_i 則代表這條道路的長度。

保證測資中任兩座山峰都有一條路徑連接這兩座山峰。

輸出格式

輸出魔法瘋狂路徑的長度，結尾必須換行。

資料範圍

- $1 \leq E \leq 1000000$
- $0 \leq a_i, b_i \leq E$ and $0 < c_i \leq 2^{31} - 1$ 對所有 $i = 1, 2, \ldots, E$

輸入範例 1

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

輸出範例 1

```
12
```

輸入範例 2

```
5
0 1 3
0 2 4
0 3 5
0 4 1
0 5 8
```

輸出範例 2

```
13
```

輸入範例 3

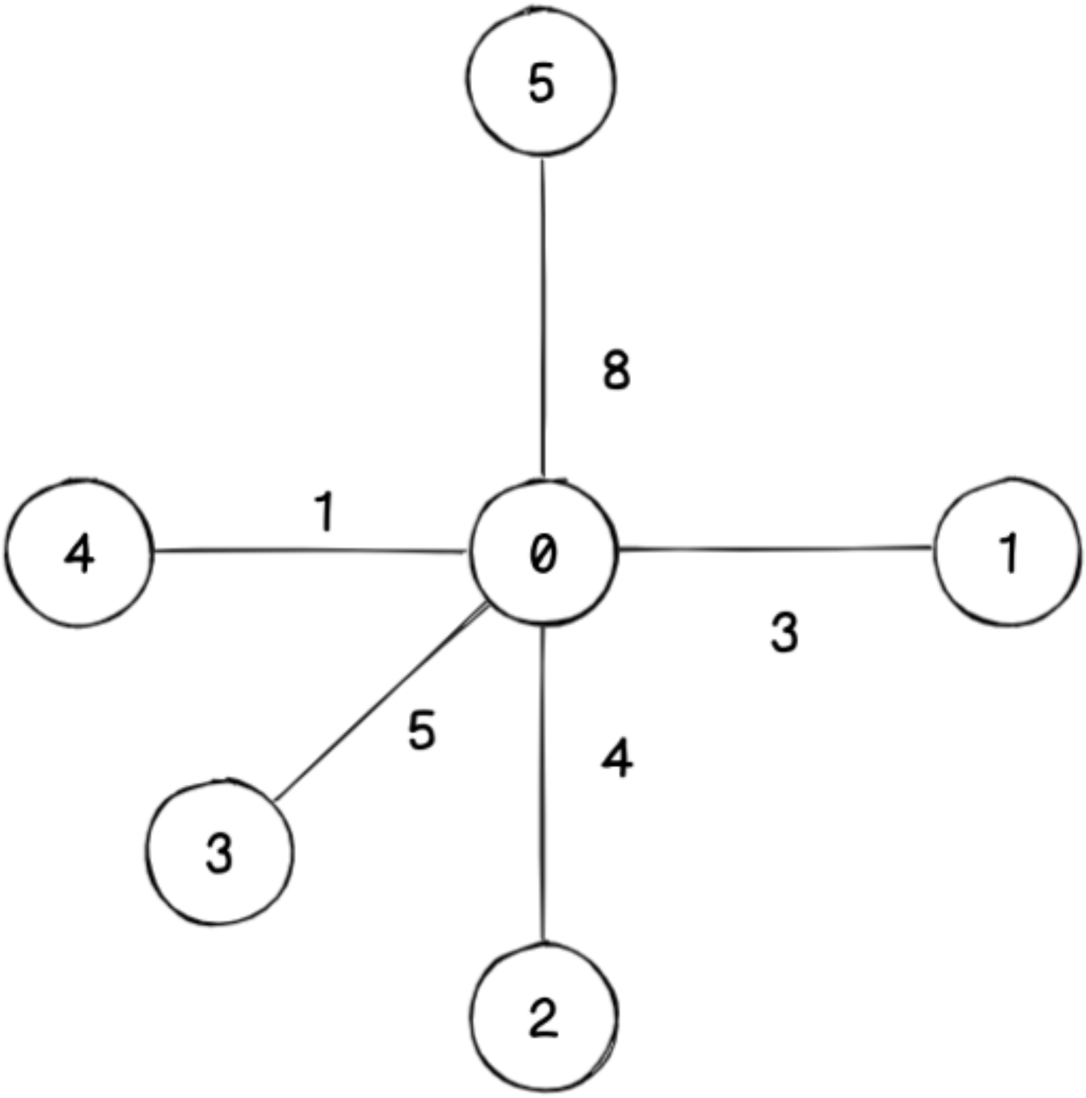
```
5
0 1 10
1 2 10
1 3 10
1 4 10
4 5 10
```

輸出範例 3

```
30
```

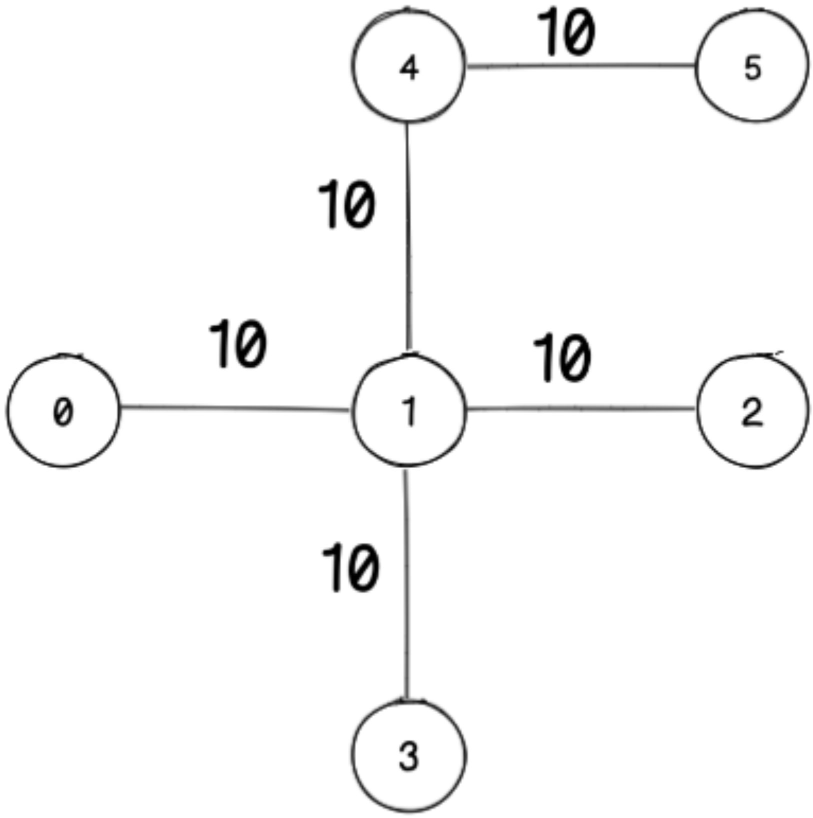
範例說明

- 範例 1. 這個例子在問題敘述中有解釋。
- 範例 2.



最長的路徑是 $(3 \to 0 \to 5)$ ，所以這個國家的魔法瘋狂距離是 13。

- 範例 3.



從山峰 0 到山峰 5，距離是 30。

10_青蛙過河（Frog Crossing River）

(25分)

時間限制：3秒
記憶體限制：512MB

問題敘述

有一隻青蛙現在在某河的一岸，而牠想要踩著河中那些露出水面的石頭，一步一步跳到另一岸去。

在這裡，我們可以把河中石頭的位置，對應到一個二維平面的整數點上。給定一顆露出水面的石頭，我們可以把其座標用(m, n)表示，其中，$1 \leq m \leq$ 河寬，$1 \leq n \leq$ 河長。

這隻青蛙過河的方法是，牠會先從所有座標滿足$m = 1$的石頭中挑一顆石頭(假設座標是$(1, n_1)$)，然後從河岸上跳到那顆石頭，$(1, n_1)$上。再來，牠會從所有座標滿足$m = 2$的石頭中挑一顆石頭(假設座標是$(2, n_2)$)，然後從$(1, n_1)$跳到$(2, n_2)$。依此類推，如果河寬為W，那麼牠最後會從$(W-1, n_{W-1})$跳到(W, n_W)，再從(W, n_W)跳到對面的河岸。

但是，不同的跳躍距離，會對青蛙造成不同程度的體力耗損。已知青蛙從最初的河岸跳到座標是$(1, n_1)$的石頭，或是從座標是(W, n_W)的石頭跳到另一邊的河岸，都是不耗體力的；但是，從座標是(i, n_i)的石頭，跳到座標是$(i+1, n_{i+1})$的石頭，會耗費$|n_i - n_{i+1}|$的體力。

請你幫這隻青蛙算算看，在給定河的長度、寬度，以及所有露出水面的石頭的座標的情況下，牠至少要耗費多少體力，才能夠跳到河的對岸？

輸入格式

- L W
- N_1 $r_{(1,1)}$ $r_{(1,2)}$ ... $r_{(1,N_1)}$
- N_2 $r_{(2,1)}$ $r_{(2,2)}$ ... $r_{(2,N_2)}$
- ...
- N_W $r_{(W,1)}$ $r_{(W,2)}$ ... $r_{(W,N_W)}$

輸入的第一行包含兩個正整數，L、W，分別代表河的長度與寬度。

接下來會有W行，其中的第i行（$1 \leq i \leq W$）一開始會有一個正整數N_i。再來會有N_i個正整數，$r_{(i,1)}$，$r_{(i,2)}$，...，$r_{(i,N_i)}$，表示在座標$(i, r_{(i,1)})$、$(i, r_{(i,2)})$、...、$(i, r_{(i,N_i)})$都有露出水面的石頭。

在這裡，不論i為何，都有$1 \leq r_{(i,1)} < r_{(i,2)} < ... < r_{(i,N_i)} \leq L$。

輸出格式

輸出只有一個整數，表示青蛙所會耗費的最小體力。最後要用一個'\n'結尾。

資料範圍

- $1 \le L \le 100000.$
- $1 \le W \le 100.$
- $\sum_{i=1}^{W} N_i \le 1000000$
- $1 \le N_i \le L \, , \forall 1 \le i \le W$

輸入範例 1

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

輸出範例 1

```
2
```

輸入範例 2

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

輸出範例 2

```
4
```

輸入範例 3

```
10 4
3 1 6 9
2 1 8
3 3 4 5
4 1 4 5 7
```

輸出範例 3

```
3
```

範例說明

在範例一中，青蛙如果沿著座標$(1,4)$，$(2,3)$，$(3,2)$的石頭跳，就只要耗費$|4-3|+|3-2|=2$點的體力。

在範例二中，青蛙如果沿著座標$(1,5)$，$(2,3)$，$(3,5)$的石頭跳，就只要耗費$|5-3|+|3-5|=4$點的體力。

在範例三中，青蛙如果沿著座標$(1,1)$，$(2,1)$，$(3,3)$，$(4,4)$的石頭跳，就只要耗費
$|1-1|+|1-3|+|3-4|=3$點的體力。