

A. $A+B=N$ (ABN)

問題敘述

相信有在打競賽的各位，都聽過 $A+B$ Problem 吧，就是給你 A 跟 B ，請你輸出 $A+B$ 的值，我們現在把這題做一點變化，給你 $A+B$ 的值 N ，請你構造出 M 組不同的 A 和 B ，使得這 M 組的 $A+B=N$ 。

不同的定義為 A 或 B 任一個不同，即視為不同解，也就是說 $3+2 \neq 2+3$ 。

請注意：你輸出的 A 和 B 必須為整數且符合 $0 \leq A, B \leq 10^9$ 。

輸入格式

N M

- N, M 的意義如題目所述。

輸出格式

A_1 B_1
 A_2 B_2
⋮
 A_M B_M

- (A_i, B_i) 為一組滿足 $A+B=N$ 的 (A, B) 。
- 若輸出超過 M 行、輸出不該輸出的字元或是輸出的數字範圍不符合題目要求，該筆測試資料將被視為 **Wrong Answer**。

測資限制

- $2 \leq N \leq 10^9$ 。
- $1 \leq M \leq \min(100, N)$ 。

範例測試

Sample Input	Sample Output
5 3	1 4 2 3 3 2
16 8	1 15 2 14 3 13 4 12 5 11 6 10 7 9 8 8

評分說明

本題共有 4 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	20	$N = 2, M = 1$ 。
2	30	$M = 1$ 。
3	50	$M = 100$ ，你在這個子任務會獲得的分數是你輸出正確的數對數量一半。

B. 重訪抽水機 (Rewater)

問題敘述

西門大地主在西門擁有 N 塊田。每塊田中有一個灑水器，灑水器內可以儲存水，灑水器之間也會有水管連通，以共用所需的水量。已知任兩個灑水器之間都存在一個水管的路徑將他們互相連接，而總共只有 $N-1$ 條水管。

每次將一塊田的灑水器啟動時，原本儲存在該灑水器的水會全部被灑出，而所有其他灑水器上的水會因為負壓而往該灑水器移動一格。

西門是個時常降雨的地方，已經連續下了 Q 天的雨。在第 i 天的時候，西門大地主發現下雨導致一共有 m_i 塊田地的灑水器有至少一單位的水，其中這些田地的編號為 $a_{i,1}, a_{i,2}, \dots, a_{i,m_i}$ ，其餘 $N - m_i$ 塊田的灑水器都是空的。然而西門大地主認為這一天第 t_i 塊田要有至少一單位的水，讓那裡的植物可以順利成長。他想要運用灑水器來達成這個目標，而為了讓生活變得有趣一點，他會反覆進行以下步驟，直到完成他的目標：

- 擲一個公正的 N 面骰，令擲出的數字為 x ，啟動位於第 x 塊地的灑水器任意**非負整數**次。

西門大地主很好奇當他使用最佳策略挑選啟動的次數時，他期望需要擲**幾次骰子**才能達成他的目標？請對於每一天都幫他計算出答案。

輸入格式

```

N
u1 v1
u2 v2
⋮
un-1 vn-1
Q
m1 t1
a1,1 a1,2 ⋯ a1,m1
m2 t2
a2,1 a2,2 ⋯ a2,m2
⋮
mQ tQ
aQ,1 aQ,2 ⋯ aQ,mQ

```

- N, Q 分別表示田地和降雨的天數的數量。
- u_i, v_i 表示第 i 條水管連接的田地編號。
- $m_i, t_i, a_{i,j}$ 的意義與題目相同。

輸出格式

```
ans1  
ans2  
⋮  
ansQ
```

- ans_i 代表第 i 天的答案。

測資限制

- $2 \leq N \leq 3 \times 10^5$
- $1 \leq Q \leq 3 \times 10^5$
- $1 \leq \sum_{i=1}^Q m_i \leq 3 \times 10^5$
- $1 \leq m_i \leq N$
- $1 \leq u_i, v_i, t_i, a_{i,j} \leq N$
- $a_{i,j} < a_{i,j+1}$

範例測試

Sample Input	Sample Output
6 1 2 2 3 2 4 1 5 4 6 3 2 3 1 5 1 4 3 6 1 1 2 3 4 5 6	6.0000000000 3.0000000000 0.0000000000
5 1 2 2 3 3 4 4 5 3 1 3 1 2 4 2 5 3 4 1 2 3	1.6666666667 1.0000000000 2.5000000000

評分說明

本題共有 1 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	10	$u_i = i, v_i = i + 1$
2	25	$m_i = 1, 1 \leq n, q \leq 2000$
3	30	$m_i = 1$
4	20	$t_i = 1$
5	15	無其他限制



C. 丟雞蛋問題 (Egg)

問題敘述

丟雞蛋問題是一個經典的題目，你現在正身處於一棟 M 層樓高的大樓，手上有 60 顆一模一樣的雞蛋，已知這些雞蛋只要在**超過 h 樓**的地方將其摔出就會破裂；不超過則不會。但由於我們目前還不知道 h 是多少，請你在摔完這些雞蛋之前，回答出 h 的值。

實作細節

你需要在首行加入 `#include "Egg.h"`，並完成以下函式：

```
long long height_limit(long long M);
```

- M 代表 h 的範圍介於 $[1, M]$ ，其中 h 為整數， $1 \leq M \leq 10^{18}$ 。
- 該函式需要在被呼叫後，回傳 h 的正確數值。

你的程式可以呼叫以下函式：

```
int is_broken(long long k);
```

- 代表你要將一顆雞蛋從高度 k 的樓層摔下去，`is_broken(k)` 會回傳 1 或 0，如果回傳 1 代表雞蛋破了；0 則代表雞蛋沒破。 k 必須是一個介於 $[1, M]$ 的整數。
- 該函式只能被呼叫 60 次，即使雞蛋沒破，你也無法將其回收。

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會被判斷為 **Accepted**。

互動範例

考慮以下的測試資料： $M = 10, h = 4$ 。

一個被評分程式判斷為 **Accepted** 的互動例子顯示如下：

評分程式端	參賽者端
呼叫 <code>height_limit(10)</code>	
回傳 0	呼叫 <code>is_broken(1)</code>
回傳 0	呼叫 <code>is_broken(2)</code>
回傳 0	呼叫 <code>is_broken(3)</code>
回傳 0	呼叫 <code>is_broken(4)</code>
回傳 1	呼叫 <code>is_broken(5)</code>
	回傳 4

評分說明

本題共有 4 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	12	$M = 2$ 。
2	20	$M = 60$ 。
3	68	無額外限制。

範例評分程式

範例評分程式以下列格式讀取輸入：

- 第一列： $M h$

其中 M, h 如題目所述。

請注意：使用自己上傳的測試資料進行測試時，沒有下面 MSG 描述的情形時你會得到 **Accepted**。如果你的程式被評為 **Accepted**，範例評分程式會輸出 Accepted: q ，其中 q 表示呼叫 `is_broken` 函式的總次數。如果你的程式被評為 **Wrong Answer**，範例評分程式會輸出 Wrong Answer: MSG，其中 MSG 格式與意義如下：

- invalid broken query: 不合法的 `is_broken` 呼叫。
- too many queries: 呼叫上述函式的總次數超過 60 次。
- incorrect height: `height_limit` 回傳的值與題目的 h 不相符。

在附件檔案中，有一個名為「Egg.zip」的壓縮檔，下載後解壓縮可以找到三個資料夾 `cpp`、`c` 和 `samples`，資料夾的意義分別為：

- `cpp`：內部包含一個檔案 `Egg.cpp`，你可以參考或修改這份程式碼，並將修改過的程式碼上傳至評測系統。你可以用檔案 `compile_cpp.sh` 或 `compile_cpp.bat` 在自己的電腦上編譯。
- `c`：內部包含一個檔案 `Egg.c`，你可以參考或修改這份程式碼，並將修改過的程式碼上傳至評測系統。你可以用檔案 `compile_c.sh` 或 `compile_c.bat` 在自己的電腦上編譯。
- `samples`：內部包含互動範例的輸入。

請不要嘗試撰寫題目指定需要函式以外的任何東西，例如自行輸入、輸出等。`grader.cpp` 與 `grader.c` 僅供參考用，並與 Judge 上的有所落差。



D. 重訪黑白雞 (Rechicken)

問題敘述

注意，此題僅限使用 C++ 語言上傳，若你使用 C 語言上傳，你將會獲得 CE。

老王的農場內有一隻黑雞、一隻白雞。在一年前，黑雞與白雞為老王下了許多黑白雞蛋，讓老王過著快樂的生活。然而現在牠們已經老了，下蛋能力大不如前，老王準備要把牠們殺了，並做成雞湯來品嚐。

黑雞與白雞當然不想要被殺掉，因此老王向牠們提出了一個遊戲，只要牠們獲勝就答應不殺了牠們。遊戲的內容如下：

首先老王會先在心裡想一棵 n 個節點的樹，之後將這棵樹的 $n - 1$ 條邊告訴黑雞。而黑雞的目標是要讓白雞也知道這棵樹的樣子，為了達成這個目標，牠可以告訴白雞一個由 0 與 1 組成的字串。最後，白雞藉由這個字串告訴老王一棵同樣為 n 個節點的樹，只要這棵樹與原本老王想的樹**同構**牠們就贏了。

黑雞與白雞當然不想要被殺掉，然而由於牠們的雞腦容量太小了，黑雞最多只能跟白雞說長度至多 300 的字串，請幫助牠們獲勝。

一棵 n 個節點的樹為一張有 $n - 1$ 條邊的連通圖。兩棵 n 個節點的樹 E_1, E_2 同構若且為若存在一個長度為 n 個排列 p_1, p_2, \dots, p_n 使得如果 $(u, v) \in E_1$ ，則 $(p_u, p_v) \in E_2$ 。

實作細節

你需要完成兩個檔案 `Black.cpp` 與 `White.cpp`。`Black.cpp` 需完成以下函數：

```
std::string Black(int n, std::vector <std::pair<int, int>> edge);
```

`White.cpp` 需完成以下函式：

```
std::vector <std::pair<int, int>> White(int len, std::string str);
```

- `Black(n, edge)` 函式的參數 n 為一個整數，代表老王所想的樹的節點數量，`edge` 為一個大小為 $n - 1$ 的 `pair` 陣列，代表老王所想的樹的邊集合。其中節點的編號位於 $[1, n]$ 。
- `Black(n, edge)` 的回傳值為一個字串，代表黑雞要告訴白雞的字串。這個字串必須由字元 0 與字元 1 組成，且字串長度不能超過 300。
- `White(len, str)` 函式的參數 `len` 為一個整數，代表黑雞傳給白雞的字串長度，`str` 為一個由字元 0 與字元 1 組成的字串，代表黑雞傳給白雞的字串，且其長度為 `len`。
- `White(len, str)` 的回傳值為一個 `pair` 陣列，代表白雞所認為樹的邊集合。
- `White(len, str)` 的回傳邊集合中，邊連接的節點需介於 $[1, n]$ 。

執行細節

你的程式將與 `manager.cpp` 一起編譯為兩個執行檔 A 與 B 並同時執行。程式將會用以下方法測試：

1. 呼叫執行檔 A 的 `Black(n, edge)` 函數 Q 次，並取得回傳值。需注意 Q 次傳入的 n 與 `edge` 未必相同！
2. 呼叫執行檔 B 的 `White(len, str)` 函數 Q 次，並取得回傳值。第 i 次的參數 `str` 為前面第 i 次呼叫 `Black(n, edge)` 時的結果。
3. 進行評分。

互動範例

[執行檔 A]：

呼叫	回傳
<code>Black(2, {(1, 2)})</code>	
	<code>0</code>
<code>Black(4, {(1, 2), (2, 3), (2, 4)})</code>	
	<code>01001011101000100</code>

[執行檔 B]：

呼叫	回傳
<code>White(1, 0)</code>	
	<code>{(1, 2)}</code>
<code>White(16, 01001011101000100)</code>	
	<code>{(1, 2), (2, 3), (2, 4)}</code>

測資限制

- $2 \leq n \leq 30$

評分說明

本題共有 2 組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	15	$n \leq 15$ 。
2	85	無額外限制。

在第 2 個子任務中，假設你在所有測資都回答正確，令 len 為所有 `Black(n, edge)` 回傳字串的長度最大值，那麼你在該筆子任務的得分為：

- 如果 $len \leq 35$ ，你會得到 85 分。
- 如果 $35 < len \leq 60$ ，你會得到的分數是 $85 - 2 \times (len - 35)$
- 如果 $60 < len \leq 150$ ，你會得到的分數是 $35 - \frac{2}{9} \times (len - 60)$
- 如果 $150 < len \leq 300$ ，你會得到的分數是 $15 - \frac{1}{10} \times (len - 150)$

範例評分程式

範例評分程式以下列格式讀取輸入：

```

Q
n1
u1,1 v1,1
u1,2 v1,2
⋮
un1-1,1 vn1-1,1
n2
u2,1 v2,1
u2,2 v2,2
⋮
un2-1,1 vn2-1,1
⋮
nQ
uQ,1 vQ,1
uQ,2 vQ,2
⋮
unQ-1,1 vnQ-1,1

```

如果你的程式被評為 Accepted，範例評分程式會輸出 Accepted: len ，其中 len 表示該筆測資中所有 **Black(n, edge)** 回傳的字串最長長度。如果你的程式被評為 Wrong Answer，範例評分程式會輸出 Wrong Answer: MSG，其中 MSG 意義如下：

- Black(): String length is too long. : 函式 **Black(n, edge)** 回傳的字串長度超過 300。
- Black(): Invalid string character. : 函式 **Black(n, edge)** 回傳的字串中包含 0、1 以外的字元。
- White(): The size of the output edge set is too big. : 函式 **White(len, str)** 回傳的邊集合大小超過 30。
- White(): The size of the output edge set is not correct. : 函式 **White(len, str)** 回傳的邊集合數量不對。
- White(): The index of the output vertex is not in range [1, n]. : 函式 **White(len, str)** 回傳的邊集合的點編號不在 $[1, n]$ 中。
- White(): The output edge set is not a tree. : 函式 **White(len, str)** 回傳的邊集合不是一棵樹。

- `White()`: The output tree is not isomorphic to the input tree. : 函式 `White(len, str)` 回傳的邊集合形成的樹與輸入的樹不同構。

在附件檔案中，有一個名為「Rechicken.zip」的壓縮檔，解壓縮後會得到的檔案有：

- `samples/01.in` : 互動範例的輸入。
- `cpp/Black.cpp` 與 `cpp/White.cpp` : 在評測系統上能正常編譯，但不一定會獲得 Accepted 的範例程式碼。你可以參考或修改這份程式碼，並將修改過的程式碼上傳至評測系統。
- `cpp/Black.h` 與 `cpp/White.h` : cms 系統上與 `compile-cpp.sh` 需要引入的標頭檔。
- `cpp/manager.cpp` : 範例評分程式。
- `cpp/compile-cpp.sh` 與 `cpp/compile-cpp.bat` : 用於編譯的 script，請視你的作業系統決定你要使用哪一個檔案。將檔案分別命名為 `Black.cpp` 與 `White.cpp` 後運行 script 即可產生執行檔。

此外，以下為 judge 上編譯的指令：

```
g++ -DEVAL -std=gnu++17 -O2 -pipe -static -s -o manager manager.cpp manager.h  
Black.cpp Black.h White.cpp White.h
```

請不要嘗試撰寫題目指定需要函式以外的任何東西，例如自行輸入、輸出等。grader.cpp 僅供參考用，並與 Judge 上的有所落差。

E. 數獨 (Sudoku)

本題為 Output Only。

問題敘述

「數獨」是一款益智遊戲，在遊戲的一開始會有一個 $n^2 \times n^2$ 的表格，其中這張表格被切成了 n^2 塊區域，每一塊區域內都有 $n \times n$ 個格子。每個格子可以是空的，也可以包含一個 $1 \sim n^2$ 的數字。一個合法的數獨同時會滿足以下條件：

1. 所有在同一列的數字皆兩兩相異。
2. 所有在同一行的數字皆兩兩相異。
3. 所有在同一子區域的數字皆兩兩相異。

下圖是兩個合法的數獨範例，兩個數獨內皆不包含空的格子：

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

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

現在，給你一些尚有一些空格子的合法數獨，請你試圖填入盡可能多的數字並保持該數獨一直都是合法的。保證給定的數獨至少存在一種填滿所有空格子的填數字方法。

本題為一 output only 的任務，並且會部分給分。你將會拿到 10 個輸入檔，說明每一個未完成數獨的樣子。對於每一個輸入檔，你應該繳交一個輸出檔，該檔案描述一組填入一些數字後的成果。對於一個合法的數獨成果，你的成績將依照填入數字的多寡做相應的評分。

輸入格式

```

n
A[1][1] A[1][2] ... A[1][n^2]
A[2][1] A[2][2] ... A[2][n^2]
⋮
A[n^2][1] A[n^2][2] ... A[n^2][n^2]

```

- n 的意義如題目所述。
- $A[i][j]$ 為格子 (i, j) 裡的數字，其中 0 代表該格子為空。

輸出格式

```

B[1][1] B[1][2] ... B[1][n^2]
B[2][1] B[2][2] ... B[2][n^2]
⋮
B[n^2][1] B[n^2][2] ... B[n^2][n^2]

```

- $B[i][j]$ 代表格子 (i, j) 填入數字後的成果。

測資限制

- $2 \leq n \leq 20$ 。
- $0 \leq A[i][j] \leq n^2$ 。
- 給定的數獨 A 是合法的，且保證至少存在一種填滿所有空格子的填數字方法。

範例測試

Sample Input	Sample Output
2	4 2 3 0
0 2 0 0	3 1 2 4
3 0 0 0	1 3 4 2
0 0 4 0	2 4 0 1
0 0 0 1	

評分說明

一個被視為「合法」的輸出檔，必須滿足以下所有條件：

- 所有在同一列的數字皆兩兩相異。
- 所有在同一行的數字皆兩兩相異。
- 所有在同一子區域的數字皆兩兩相異。
- 原本不是 0 的數字依舊保持原樣。
- $0 \leq B[i][j] \leq n^2$

注意，假設你填出了一個不完整的數獨成果，你並不需要保證存在一種方法可以使剩下的空格子能夠被合法地填滿。

對於每一個合法的輸出檔，你最高可以得到 10 分。令 A 中空格子的數量為 p ，令 B 中空格子的數量為 q ，那麼你將根據以下規則得分：

- 得 $10 \times (p - q)/p$ 分。

詳見下表。本題共有 10 個測試資料檔案，條件限制如下所示。

測試資料	分數	n	額外輸入說明
1	10	2	輸入範例 1。
2	10	3	無其他限制。
3	10	3	無其他限制。
4	10	10	無其他限制。
5	10	20	$A[i][j] = 0$ 。
6	10	4	無其他限制。
7	10	8	無其他限制。
8	10	12	無其他限制。
9	10	16	無其他限制。
10	10	20	無其他限制。