

## pA.水題

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Angus 今天一早醒來，覺得自己好像喝咖啡、吃甜食、又用力，因此感到口乾舌燥，急需補充水分。

他找到了他的水壺，其容量為  $c$ ，但發現裡面連一滴水都不剩，因此衝到了廁所去裝水喝。

因為 Angus 家裡非常有錢，因此廁所裝有  $n$  個水龍頭，但為了節約用水，Angus 一開始只會打開一個水龍頭。

隨著時間流逝，他如果覺得一個水龍頭裝水太慢，則會選擇再加開一個水龍頭，避免自己渴死。

對於第  $i$  個水龍頭，會在第  $t_i$  秒打開，每秒提供  $a_i$  的水量。

請問 Angus 何時才能喝到  $c$  單位的水？(若答案不是整數，則向上取整)

註：Angus 的舌頭很長，可以伸到水壺底部，因此當水壺裝完水後不會有喝不到底部的水的問題

### Input

第一行有兩個整數  $n, c$ ，分別代表水龍頭的個數以及水壺的容量。

接下來有  $n$  行，每行有兩個整數  $t_i, a_i$ ，代表第  $i$  個水龍頭在  $t_i$  時刻開啟，每秒輸出  $a_i$  的水量。

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq c \leq 10^9$
- $1 \leq a_i \leq 10^9$
- $t_1 = 1$  且  $1 \leq t_i \leq 10^9$  且對於所有  $1 \leq j \leq n - 1$  有  $t_j \leq t_{j+1}$

### Output

Angus 至少需要多少時間才能喝到  $c$  單位的水，若答案不為整數則取大於該數的最小整數。

### Scoring

| 子任務 | 分數  | 額外測資限制 |
|-----|-----|--------|
| 1   | 100 | 無其他限制  |

### Examples

| input                                     |
|---|
| 5 100<br>1 8<br>2 2<br>4 5<br>6 9<br>10 6 |
| output                                    |
|   |

7

**input**

1 100

1 1

**output**

100

# pB.電學大師

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

有學過電學的人一定知道，只要兩點有電位差存在，並且兩點間的電阻並非無限大，那麼其之間就可以形成電流。

現在有一張圖，Caido 和 Same 在圖上玩，最初這一張圖有  $n$  個點，但彼此沒有連接。接著他們會進行總共  $q$  次指令，每次操作會輸入三個整數  $cmd, x, y$ ，分別代表指令種類與執行指令的兩個點，指令有以下兩種：

- 指令 0，Caido 站在點  $x$  上，Same 站在點  $y$  上，由於他們兩個有電位差，因此只要  $x, y$  兩點間有直接或間接連接，那麼就會通電，你需要輸出是否有通電。
- 指令 1，兩人將點  $x$  與點  $y$  連接起來，不需輸出任何內容。

## Input

輸入第一行有兩個正整數  $n, q$ ，分別代表點的數量、指令數量。

接下來有  $q$  行，每一行有三個整數  $cmd, x, y$ ，分別代表指令種類與執行指令的兩個點，指令之意義如題目所述。

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 2 \times 10^5$
- $1 \leq x, y \leq n$
- 保證同一筆指令中  $x \neq y$
- $cmd \in \{0, 1\}$

## Output

對於每筆指令 0，如果兩人間會通電輸出「YES」，否則輸出「NO」(皆不含引號)。

## Scoring

| 子任務 | 分數 | 額外測資限制             |
|-----|----|--------------------|
| 1   | 20 | $n \leq 1000$      |
| 2   | 20 | 保證指令 0 出現後不會再有指令 1 |
| 3   | 60 | 無其他限制              |

## Example

| input |
|-------|
| 5 10  |
| 0 1 2 |
| 1 1 2 |
| 0 1 2 |
| 0 1 3 |
| 1 2 3 |
| 0 1 3 |
| 1 4 5 |

|                               |
|-------------------------------|
| 1 4 3<br>1 5 1<br>0 1 4       |
| <b>output</b>                 |
| NO<br>YES<br>NO<br>YES<br>YES |

**Note**

範例測資1說明：

進行第一個指令時，任兩點皆未連通，而第二筆指令連接點 1,2，進行第三個指令時，點 1,2 間已經接通。執行第五個指令連接點 2,3，因此執行第六比指令時點 1,3 間已經接通。

## pC.臨時抱佛腳

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

明天就是段考了，但 Samuel 突然發現他還沒讀書。

這次段考範圍 Samuel 需要讀的有  $n$  本書，共有  $m$  種科目。並且這  $m$  種科目編號為  $1, 2, \dots, m$ 。Samuel 對他們分別有喜好程度  $fav_1, fav_2, \dots, fav_m$ ，喜好程度越高代表越喜歡。

這  $n$  本書編號為  $1, 2, \dots, n$ ，皆屬於  $m$  種科目之一，每本書的科目分別為  $sub_1, sub_2, \dots, sub_n$ ，另外每本書分別有難度  $dif_1, dif_2, \dots, dif_n$ 。

因為 Samuel 的時間所剩不多了，所以他決定利用以下順序讀書：

- 讀書的順序必須符合他對科目的喜好，先讀的書的科目喜好程度的順序必須大於等於後讀的書。換句話說，對於所有  $i < j$ ，讀的第  $i$  本書科目為  $x$ ，第  $j$  本書科目為  $y$ ，必須有  $fav_x \geq fav_y$ 。
- 讀書的順序必須符合書的難度，難度順序必須是嚴格遞增或嚴格遞減(因為他覺得越讀越難會比較有挑戰性，而越讀越簡單則比較輕鬆)。換句話說，對於所有  $i < j$ ，讀的第  $i$  本書與第  $j$  本書必須全部滿足  $dif_i < dif_j$  或全部滿足  $dif_i > dif_j$ 。

現在 Samuel 想要請你幫他計算他最多可以讀幾本書？

### Input

輸入有四行。第一行有兩個正整數  $n, m$ ，分別代表要讀的書的數量、科目數量。

第二行有  $m$  個正整數  $fav_1, fav_2, \dots, fav_m$ ，代表對每個科目的喜愛程度。

第三行有  $n$  個正整數  $sub_1, sub_2, \dots, sub_n$ ，代表每本書的科目。

第四行有  $n$  個正整數  $dif_1, dif_2, \dots, dif_n$ ，代表每本書的難度。

- $1 \leq m \leq n \leq 2 \times 10^5$
- $1 \leq sub_i \leq m$
- $1 \leq fav_i, dif_i \leq 10^9$

### Output

輸出一個正整數，代表在題目所述的條件下，Samuel 最多可以讀幾本書。

### Scoring

| 子任務 | 分數 | 額外測資限制        |
|-----|----|---------------|
| 1   | 12 | $m = 1$       |
| 2   | 26 | $n \leq 1000$ |
| 3   | 62 | 無其他限制         |

### Examples

| input                                  |
|--|
| 5 3<br>5 3 4<br>1 1 2 3 3<br>5 2 6 3 8 |
| output                                 |
| 3                                      |

| input  |
|--|
| 8 4<br>5 3 4 5<br>1 1 2 3 3 4 4 4<br>5 2 6 3 8 6 7 5 |
| output   |
| 5  |

## Note

範例測資1說明：

讀書的順序可以是 2, 1, 3，其中書本 2 為科目 1、難度 2，書本 1 為科目 1、難度 5，書本 3 為科目 2、難度 6，因為科目 1 的喜好程度大於科目 2，因此先讀科目 1 再讀科目 2 是可以的，另外難度的排序為嚴格遞增。可以證明沒有可以讀更多書的方法。

範例測資2說明：

讀書的順序可以是 2, 1, 6, 7, 5，因為科目 4 的喜好程度等於科目 1，因此這兩科的先後順序並沒有限制，另外難度的排序為嚴格遞增。可以證明沒有可以讀更多書的方法。

## pD.油漆與底線

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

AOG 家裡最近在裝修，因此家裡有非常多油漆，但是AOG非常調皮，喜歡拿油漆到處刷。

爸媽也拿他沒有辦法，只好亮出自己的底線，讓AOG不要犯。

其中地板只有一排，並且一共有  $len$  塊磁磚，呈直線排列，所有磁磚的初始顏色皆為 1。

因為爸媽的喜好無常，因此底線有可能一直變化。

其中一條底線以兩個數字  $x$  和  $c$  的方式呈現，代表父母當下希望第  $x$  格必須是顏色  $c$ 。

調皮的 AOG 用油漆刷了多次地板，其中第某次用顏色  $c$  的油漆從第  $l$  格磁磚刷到第  $r$  格磁磚，而原本的顏色會被後來刷上的顏色覆蓋。

父母有可能在 AOG 刷地板的過程中提出底線，而他們講完之後就會忘記自己之前的底線。

因此只要在得到底線資訊的當下，判斷AOG有沒有觸碰到爸媽的底線即可。

因為爸媽很健忘，因此就算後續操作中違反了底線也沒關係。

請注意，由於題目是詢問是否觸碰到父母底線，因此若是目前該塊磁磚顏色和父母希望的顏色不同，請輸出 YES。

### Input

第一行有兩個整數  $len, n$ ，分別代表磁磚數量與指令數量。

接下來有  $n$  行輸入，共有兩種指令：

第一種為「1  $l$   $r$   $c$ 」，代表 AOG 以顏色  $c$  的油漆從第  $l$  格磁磚刷到了第  $r$  格磁磚。

第二種為「2  $x$   $c$ 」，代表父母當下希望第  $x$  格磁磚的顏色為  $c$ 。

- $1 \leq len, n \leq 10^6$
- $1 \leq l, r, x \leq len$
- $1 \leq c \leq 10^9$

### Output

在每次獲得一個新的底線的當下，輸出 AOG 是否會觸碰到爸媽的底線。若會，輸出「YES」；反之，則輸出「NO」。

### Scoring

| 子任務 | 分數 | 額外測資限制             |
|-----|----|--------------------|
| 1   | 27 | $len, n \leq 1000$ |
| 2   | 73 | 無其他限制              |

Example

| input  |
|--|
| 8 5<br>2 6 1<br>1 5 8 3<br>2 6 3<br>1 1 5 2<br>2 5 2 |
| output   |
| NO<br>NO<br>NO                                       |



# pE.下棋

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

小石平時非常喜歡下棋，無論是甚麼棋他都相當精通，今天他的朋友小謙向他介紹了一種新的棋。

棋盤為二維座標平面上的一個矩形，而其中遊戲規則如下：

- 棋子只能下在一個左下角座標  $(a, b)$  且右上角座標  $(c, d)$  的矩形內。
- 任兩顆棋子不能重疊。
- 一人一回合只能下一顆棋。

由於小謙知道小石很強，如果沒有限制小石發揮的話自己一定會輸。

因此若小石在  $(x, y)$  的地方下了棋，則需付給小謙  $\gcd(x, y)$  元，而小謙下棋則不需要任何花費。

但小謙又覺得這樣不太公平，因此每局都讓小石先手，自己後手。

下棋的雙方都要盡可能的下棋，越多越好。

今天小石出門的時候帶了  $k$  元，設雙方皆以最佳策略下棋，試問小石最多可以下多少棋。

## Input

第一行有四個整數  $a, b, c, d$ ，代表棋盤的左下角  $(a, b)$  和右上角  $(c, d)$ 。第二行會有一個數字  $k$ ，代表小石今天帶了多少錢。

- $1 \leq a \leq c \leq 10^7$
- $1 \leq b \leq d \leq 10^7$
- $1 \leq k \leq 10^{15}$

## Output

輸出一個整數  $n$ ，代表小石最多可以下多少棋。

## Scoring

| 子任務 | 分數 | 額外測資限制                 |
|-----|----|------------------------|
| 1   | 36 | $a, b, c, d \leq 1000$ |
| 2   | 64 | 無其他限制                  |

## Examples

|                |
|----------------|
| input          |
| 2 3 8 10<br>15 |
| output         |
|                |

**input**

10 10 20 20

10000000000000000

**output**

61

# pF.跑腿大師

time limit per test: 4 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

在 AOG 的夢中，他回到了那個「媽媽十塊」的年代，為了拿到零用錢，他只能幫媽媽跑腿。

為了協助他賺取零用錢，你必須先知道以下幾件事。

首先，AOG 跑腿的目的地和其之間的路能表示為一棵樹，這棵樹上有  $n$  個點，與  $n - 1$  條邊，AOG 必須沿著路(也就是邊)在目的地(也就是點)之間穿梭。

再來，他有  $q$  個跑腿任務，每個任務有兩個地點  $l, r$  (保證  $l < r$ )，這並不代表他只需要從  $l$  跑到  $r$ ，零用錢可沒這麼好賺，他必須從  $l$  依序前往  $l + 1, l + 2, \dots, r$  才可以拿到零用錢。

最後，畢竟他這麼辛苦，媽媽決定多給他一點零用錢，數量取決於路徑  $l, l + 1, l + 2, \dots, r$  的距離，但畢竟要去的地方真的太多了，AOG 也沒心思多繞路來賺零用錢，因此他全程只會走最短的路線。另外因為媽媽擔心他走太久，所以給他準備了一個背包裝滿小點心，而背包內可以裝下  $k$  個小點心，AOG 每走一單位的距離就會吃掉一個小點心，不過每到達路徑上的一個目的地，AOG 又可以在當地把小點心給補滿(假如目前在從  $i$  到  $i + 1$  的路上，那麼必須走到  $i + 1$  才能補給小點心，不能在半路進行補給)，但倘若在半路上沒有小點心了，他就會走不動而無法完成任務(為了簡化狀況，我們令兩地最短距離為  $d$ ，則背包空間必須滿足  $k \geq d$ )。

現在 AOG 正準備要出發，想要請你幫忙計算每個跑腿任務走玩  $l, l + 1, l + 2, \dots, r$  的距離，也就是他可以賺的零用錢，以及對於每個跑腿任務它至少要帶空間多大的背包，才不會半路沒有小點心。

## Input

輸入第一行有兩個正整數  $n, q$ ，分別代表點的數量與跑腿任務的數量。

接下來有  $n - 1$  行，每行有三個正整數  $u, v, dis$ ，代表其中一條路連接  $u$  與  $v$  並且長度為  $dis$ 。

接下來有  $q$  行，每行有兩個正整數  $x, y$ ，代表其中一個跑腿任務需要從  $x$  沿途經過  $x + 1, x + 2, \dots, y$ 。

- $2 \leq n \leq 2 \times 10^5$
- $1 \leq q \leq 2 \times 10^5$
- $1 \leq u, v \leq n$
- $1 \leq dis \leq 10^6$
- $1 \leq x < y \leq n$

## Output

輸出  $q$  行，每行有兩個整數，代表對於每個跑腿任務，可以賺到的零用錢以及需要攜帶的背包大小。

## Scoring

| 子任務 | 分數 | 額外測資限制  |
|-----|----|---|
| 1   | 12 | $q \leq 10$                                     |
| 2   | 19 | 保證對於 $1 \leq i \leq n - 1$ 有一條邊連接 $i$ 與 $i + 1$ |
| 3   | 19 | $n \leq 200$                                    |

|   |    |       |
|---|----|-------|
| 4 | 50 | 無其他限制 |
|---|----|-------|

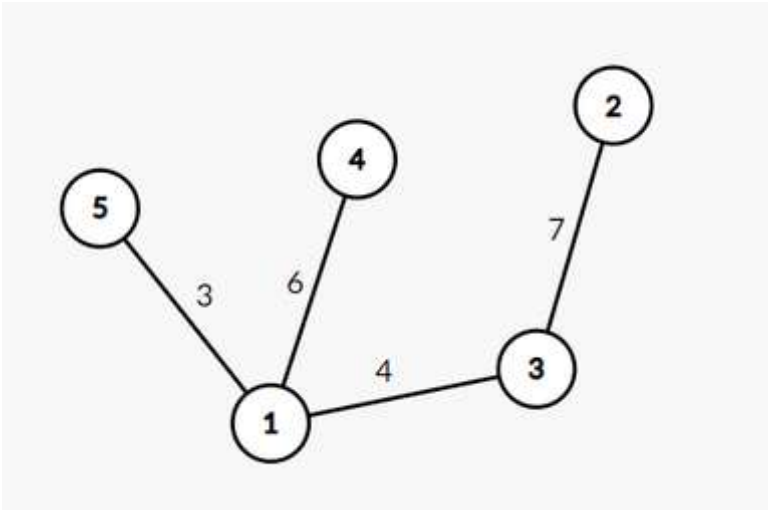
Example

| input  |
|--|
| 5 3<br>1 3 4<br>2 3 7<br>5 1 3<br>1 4 6<br>2 3<br>1 3<br>1 5 |
| output   |
| 7 7<br>18 11<br>37 11  |

Note

範例測資1說明：

各地點之間的關係如下。



對於第一筆查詢，路徑 2,3 可以直接從點 2 走到點 3，距離為 7，因此路徑總距離為 7 且需要攜帶容量為 7 的背包；對於第二筆查詢，路徑 1,2,3 需先從點 1 走到點 2，途中會經過點 3，該段路的距離是  $4 + 7 = 11$ ，再走到點 3，該段路的距離是 7，因此路徑總距離為 17 且需要攜帶容量為 11 的背包

# pG.香彤愛薩梅

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

香彤 (Same) 是個就讀臺南一中數資班的少女，她暗戀班上一位叫做薩梅 (SaMe) 的同學，於是 Same 決定要搭訕 SaMe。

「嗨哈囉哈囉 (發出氣喘笑聲)，我剛看到你覺得你還蠻可愛的，想認識你一下。」

SaMe 覺得 Same 好奇怪，於是他給 Same 一棵樹，這棵樹有  $n$  個點，編號  $i$  的點有一個點權  $a_i$ ，以 1 為根，Same 要對  $i = 1 \sim n$  回答問題：以  $i$  為根的子樹要分成  $k$  組，每組數量相同，且同一組內的點權都必須相同，問有幾種不同的  $k$  使得可以成功分組。

輸出完這  $n$  個答案後，會有  $q$  個操作：

- 第 1 種操作會給  $p, x$ ，代表要將  $a_p$  改成  $x$ 。
- 第 2 種操作會給  $p$ ，回答做完前面的操作後編號為  $p$  的子樹有幾種合法的  $k$ 。

## Input

第一行有兩個整數  $n, q$ ，代表樹的節點數量與操作次數。

第二行有  $n$  個正整數  $a_1 \sim a_n$ ，代表每個節點的點權。

接下來  $n - 1$  行，每行有兩個正整數  $u, v$ ，代表編號  $u, v$  的節點在樹上以一條邊連接。

接下來  $q$  行，每行的第一個數字是一個正整數  $t$ ，代表操作類別：

若  $t = 1$ ，會接著輸入兩個正整數  $p, x$ ，代表要將  $a_p$  改成  $x$ 。

若  $t = 2$ ，會接著輸入一個正整數  $p$ ，代表要回答以  $p$  為根的子樹有幾個不同的  $k$  使得可以成功將這個子樹分成  $k$  組。

- $1 \leq n \leq 2 \times 10^5$
- $0 \leq q \leq 2 \times 10^5$
- $1 \leq a_i \leq n$
- $1 \leq u, v \leq n$
- $1 \leq t \leq 2$
- $1 \leq p, x \leq n$

## Output

一開始先輸出一行  $n$  個整數，第  $i$  個整數代表初始時以  $i$  為根的子樹有幾種合法的  $k$ 。

接下來則對每個  $t = 2$  的操作輸出一行一個整數代表答案。

## Scoring

| 子任務 | 分數 | 額外測資限制 |
|-----|----|--------|
|-----|----|--------|

|   |    |                  |
|---|----|------------------|
| 1 | 13 | $n, q \leq 300$  |
| 2 | 17 | $n, q \leq 2000$ |
| 3 | 25 | $q = 0$          |
| 4 | 45 | 無其他限制            |

### Examples

| input  |
|--|
| 5 6<br>2 3 2 3 2<br>1 4<br>2 4<br>3 1<br>5 2<br>1 4 2<br>2 4<br>1 2 2<br>2 2<br>1 3 5<br>2 3 |
| output   |
| 1 1 1 1 1<br>1<br>2<br>1   |

| input  |
|--|
| 10 0<br>3 3 3 3 3 3 3 3 3 3<br>2 5<br>9 6<br>4 7<br>3 8<br>1 5<br>2 9<br>6 8<br>3 10<br>4 10 |
| output   |
| 4 4 3 2 3 4 1 2 2 2  |

# pH.章程和漢堡

time limit per test: 5 seconds

memory limit per test: 1024 megabytes

input: standard input

output: standard output

怡漾是闔家歡漢堡的一個店員，身為一個員工，她必須符合這家店的章程。

可以把菜單想成是一個大小為  $n \times m$  的表格，表格第  $i$  行第  $j$  列的元素是一個美味度為  $a_{i,j}$ 、價錢為  $w_{i,j}$  的漢堡包。

有  $q$  個客人要跟怡漾點餐，每個客人會告訴怡漾 6 個正整數  $u, d, l, r, x, y$ ，代表這位客人想要在菜單上第  $u \sim d$  行的第  $l \sim r$  列這個區間的漢堡包，選美味度第  $x$  小到第  $y$  小的漢堡包來吃，若有多種選擇方法，客人會選花的錢最少的方式。為了讓怡漾遵守闔家歡漢堡的章程，對於每個客人的點餐，請你幫幫怡漾回答這位客人吃漢堡包的美味度總和與價錢總和。

因為客人要依序點餐，所以怡漾必須要按順序處理客人的點餐，假設上一位客人的美味度總和與最小價錢總和是  $ans$  和  $wns$  (若輪到第一位客人點餐則  $ans = wns = 0$ )，那輪到目前的客人時會輸入的是  $u', d', l', r', x', y'$ ，要用  $ans, wns$  推算出  $u, d, l, r, x, y$  是什麼：

- $area = (d - u + 1) \times (r - l + 1)$
- $u = ((u' + ans) \bmod n) + 1, d = ((d' + ans) \bmod n) + 1$
- $l = ((l' + ans) \bmod m) + 1, r = ((r' + ans) \bmod m) + 1$
- $x = ((x' + wns) \bmod area) + 1, y = ((y' + wns) \bmod area) + 1$

## Input

第一行有三個正整數  $n, m, q$ ，代表菜單的大小與客人的數量。

接下來  $n$  行，第  $i$  行有  $m$  個正整數  $a_{i,1} \sim a_{i,m}$ ，代表漢堡包的美味度。

接下來  $n$  行，第  $i$  行有  $m$  個正整數  $w_{i,1} \sim w_{i,m}$ ，代表漢堡包的價錢。

接下來  $q$  行，每行有六個整數  $u', d', l', r', x', y'$ ，可用這六個整數和上一輪的答案推算出  $u, d, l, r, x, y$ ，代表這位客人想吃菜單上第  $u \sim d$  行的第  $l \sim r$  列美味度第  $x$  小到第  $y$  小的漢堡包。

- $1 \leq n \times m \leq 2 \times 10^4$
- $1 \leq q \leq 5 \times 10^5$
- $1 \leq a_{i,j}, w_{i,j} \leq 10^9$
- $0 \leq u', d' < n$
- 保證推算後滿足  $1 \leq u \leq d \leq n$
- $0 \leq l', r' < m$
- 保證推算後滿足  $1 \leq l \leq r \leq m$
- $0 \leq x', y' < (d - u + 1) \times (r - l + 1)$
- 保證推算後滿足  $1 \leq x \leq y \leq (d - u + 1) \times (r - l + 1)$

## Output

對於每個客人的點餐，你要回答這位客人吃的漢堡包的美味度總和與最小價錢總和。

Scoring

| 子任務 | 分數 | 額外測資限制           |
|-----|----|------------------|
| 1   | 10 | $q \leq 100$     |
| 2   | 12 | $n = 1, a_i$ 皆相異 |
| 3   | 13 | $n = 1$          |
| 4   | 13 | $a_i$ 皆相異        |
| 5   | 52 | 無其他限制            |

Examples

| input   |
|---|
| 1 6 2<br>1 1 1 2 2 2<br>1 2 4 8 16 32<br>0 0 1 4 1 2<br>0 0 4 1 3 1 |
| output  |
| 3 10<br>5 26  |

| input   |
|---|
| 3 3 1<br>1 1 1<br>1 1 1<br>1 1 1<br>1 2 4<br>8 16 32<br>64 128 256<br>1 2 0 2 3 5 |
| output  |
| 3 56  |