差分陣列

說明

給定一個整數 n，代表一個陣列 A 的長度，一開始所有數字都是 0。 接下來會有 m 筆操作，每一筆操作會把陣列中從位置 a 到 b（包含 a 和 b）的數字全部加 1。 請你使用「差分陣列」的方法處理所有操作。

差分陣列是什麼？

當你要對陣列做「很多次區間加法」的時候（像是把第 2 到第 5 個數字全部加 1），如果每次都一個一個加，速度會很慢。這時候就可以用「差分陣列」這個方法，它有以下的好處：

- 每次操作 只需要動兩個位置，非常快速
- 所有操作做完之後，只要「一路累加」，就能還原出最後的結果
- 效率很高，特別是操作很多的時候

假設你有原本的陣列 A = [0, 0, 0, 0, 0]
我們建立一個差分陣列 D，初始也都是 0。
每次對區間 [a, b] 加 1，就：

- D[a] += 1
- D[b + 1] -= 1（前提是 b + 1 還在陣列範圍內）

這樣就能表示整段都加了 1。
最後把 D 從左到右一路加總，就能得到真正的 A。

輸入

(1) 兩個整數 n 和 m
(2) m 行，每行輸入兩個整數 a 和 b

輸出

(1) 差分陣列的內容
(2) 陣列 A 最後的內容（經過所有操作之後）

範例

輸入：

5 3

0 2

1 3

2 4

輸出：

1 1 1 -1 -1

1 2 3 2 1

說明：

一開始 A 是 [0, 0, 0, 0, 0]

差分陣列 D 也是 [0, 0, 0, 0, 0]

- 操作[0,2] → **D[0] += 1, D[3] -= 1** → D = [1,0,0,-1,0]
- 操作 [1,3] → **D[1] += 1, D[4] -= 1** → D = [1,1,0,-1,-1]
- 操作 [2,4] → **D[2] += 1** → D = [1,1,1,-1,-1]

最後我們從 D 陣列還原出 A 的過程如下（前綴和）：

A[0] = D[0] = 1

A[1] = A[0] + D[1] = 1 + 1 = 2

A[2] = A[1] + D[2] = 2 + 1 = 3

A[3] = A[2] + D[3] = 3 + (-1) = 2

A[4] = A[3] + D[4] = 2 + (-1) = 1

限制

(1) 1 ≤ n,m ≤ 1000

(2) 0 ≤ a ≤ b < n

(3) 執行時間 2 秒內