

C. 芽芽想爬分

Description

天瘋是一款相當受歡迎的線上對戰遊戲，遊戲的競技水準之高在同類型遊戲中數一數二，也因此天瘋的遊戲積分被玩家們視為個人實力的重要指標；在積分榜上登頂、達到最高積分上限的玩家甚至能在遊戲官方網頁上被公開表揚，千古留名。天瘋玩家們整天整天的、廢寢忘食地內卷爬積分，有人掉分掉到破防，有人整天玩遊戲導致睡眠不足，誰看了都得說一句「天啊，又瘋了一個」，天瘋因此而得名。

不幸的是，芽芽也迷上天瘋了！但芽芽可是個成熟理智的玩家，自然會謹慎地規劃體力、調整心態，以求最高效率的賺積分。芽芽根據他蒐集的情報，已經總結出天瘋的積分計算規則了：在天瘋遊戲內每一場遊戲有四名玩家，分別獲得第一到四名，一場遊戲結束後會根據玩家現在的積分 r 和他這一場遊戲的名次 k ，獲得 $d_{r,k}$ 的積分變化。不過，玩遊戲也是消耗體力的，芽芽可以花很大的力氣認真玩，也可以隨便玩玩當休息；他知道如果他現在的積分是 r ，玩了一場遊戲得到第 k 名，那他的疲勞感會增加 $e_{r,k}$ 單位。積分是一個上限是 N 的非負整數，遊戲結束後積分超過 N 的話會直接變成 N ，而變成負數的話芽芽會心理崩潰而無法繼續進行遊戲，這是絕對不可以發生的。疲勞感倒是沒有上限也沒有下限。

在接下來 Q 天內的第 i 天，芽芽會從 s_i 積分開始，目標是在這天結束時積分至少在 t_i 或以上。他在這一天內可以在不崩潰的前提下（積分隨時大於或等於零）進行任意多場遊戲，並且希望能讓增加的疲勞感最少。不過，也是有可能芽芽不論怎麼玩都不可能爬上至少 t_i 積分。能不能請你幫芽芽規劃規劃，接下來的每一天至少需要讓疲勞感增加多少才能達成芽芽的積分目標、或者是回報芽芽不可能達成目標？

Input

輸入的第一行有兩個以空白分隔的整數 N 、 Q 。接下來的 $N + 1$ 行，第 i 行有八個以空白分隔的整數 $d_{i,1}$ 、 $e_{i,1}$ 、 $d_{i,2}$ 、 $e_{i,2}$ 、 $d_{i,3}$ 、 $e_{i,3}$ 、 $d_{i,4}$ 、 $e_{i,4}$ 。再接下來的 Q 行，第 i 行有兩個以空白分隔的整數 s_i 、 t_i 。

- $1 \leq N \leq 3000$
- $1 \leq Q \leq 10^6$
- $|d_{i,j}| \leq N$
- $|e_{i,j}| \leq 10^5$
- $0 \leq s_i, t_i \leq N$
- 不可能存在一種玩遊戲的方式，使得芽芽的疲勞度無限制的降低

Output

對於接下來 Q 天的每一天：如果芽芽可以達成目標的話，請輸出一行一個整數，代表芽芽為了達成積分目標，這一整天的疲勞感總共最少要增加多少單位；否則，如果芽芽無法達成目標，請輸出一行一個字串「. _ .」。

Sample 1

Input	Output
4 5	-10
3 20 1 10 -1 -10 -3 -20	0
2 30 1 20 -1 -10 -2 -30	10
2 20 1 10 -1 -10 -2 -20	10
3 20 2 10 -2 0 -3 0	20
0 40 -1 30 -2 20 -3 10	
1 0	
1 1	
1 2	
1 3	
1 4	

以下是每一天芽芽的最佳遊戲策略之一。在第三天時，芽芽可以只玩一場遊戲獲得第二名，結束時積分是 2，增加 20 單位疲勞感，但因為芽芽的目標是最終積分「至少」是 2，而達到 3 積分需要的疲勞感更低，所以如此才是更好的策略。第五天時，芽芽的積分超出上限 4 但沒有關係，積分會被限制在最大值。

	遊戲前積分	遊戲名次	遊戲後積分	疲勞感
第一天	1	3	0 (-1)	-10
第二天	1	3	0 (-1)	-10
	0	2	1 (+1)	±0
第三、四天	1	3	0 (-1)	-10
	0	1	3 (+3)	+10
第五天	1	3	0 (-1)	-10
	0	1	3 (+3)	+10
	3	2	4 (+2)	+20

Sample 2

Input	Output
1 2	0
-1 -100 -1 -200 -1 -300 -1 -400	· _ ·
0 0 0 0 0 0 0 0	
0 0	
0 1	

第一天，芽芽只要開始遊戲，他的積分就會掉到零以下，但他永遠不可以讓自己的積分小

於零；他的最佳策略、同時也是唯一可行的策略是不參加任何一場遊戲。第二天，芽芽沒有任何辦法獲得至少 1 積分，因此輸出「._.」。

Sample 3

Input	Output
10 10	6
0 18 7 6 4 9 7 16	11
7 11 8 8 2 18 -1 20	8
3 18 8 14 6 12 7 12	11
-3 11 2 17 6 11 2 13	24
-2 20 4 9 1 17 2 20	2
-1 1 -3 15 -3 -1 -1 -11	12
3 14 -5 7 -5 19 2 10	30
1 5 -5 16 -5 10 -5 6	0
-5 15 -3 19 0 8 -8 19	11
-7 3 -2 1 -4 5 0 14	
0 3 -9 6 -6 18 -2 10	
2 3	
3 5	
1 7	
3 6	
0 9	
3 4	
2 6	
4 10	
9 9	
0 8	

此範例滿足第一、四、五子任務的限制。

Sample 4

Input	Output
10 10	3
7 12 9 19 2 1 1 10	6
8 6 4 15 8 3 10 16	0
8 16 8 19 4 0 5 1	0
10 4 7 17 9 11 4 3	1
4 4 4 16 6 2 5 8	3
5 19 5 17 4 12 5 20	1
2 12 3 14 4 2 8 1	11
6 11 7 17 7 16 6 16	1
7 17 7 3 2 12 10 1	3
7 10 8 14 4 11 4 3	
7 7 4 17 2 4 9 20	
3 4	
1 10	
4 0	
3 3	
0 1	
3 4	
0 2	
7 9	
2 9	
1 9	

此範例滿足所有子任務的限制。

配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Input 所描述的範圍。

子任務編號	子任務配分	測試資料範圍
1	0%	範例測試資料
2	20%	$d_{r,k} > 0$ ($0 \leq r \leq N, 1 \leq k \leq 4$)
3	20%	$e_{r,k} \geq 0$ ($0 \leq r \leq N, 1 \leq k \leq 4$)
4	20%	$N \leq 500$
5	40%	無特別限制

Hint 1

本題測試資料量大，建議使用 `scanf` 進行輸入。若使用 `std::cin` 輸入，請在 `main` 函式第一行加上 `ios_base::sync_with_stdio(0); cin.tie(0);`，且請勿跟 `scanf` 混用，以免造成 `Time Limit Exceeded`。