

題目 536

536. 暗粒子

[Submit](#)[Status](#)[Discussion / Solution](#)[Ranklist](#)[Back to Problems List](#)**TopCoder****Caido**

主唱太拼命了

User's AC Ratio

66.7% (8/12)

Submission's AC Ratio

22.2% (10/45)

Tags[Show solution-related tags](#)

apcsc apcsc-loop

Description

你最近在研究一種從暗物質中萃取出來的新粒子－暗粒子，為了更加的深入研究，你決定利用程式模擬在一個大小為 $N \times M$ 的盒子中，兩顆一模一樣的暗粒子在二維方向上的移動軌跡。

一開始你會設定兩顆暗粒子的初始座標 (x_1, y_1) 以及 (x_2, y_2) 、以及初始速度 (dx_1, dy_1) 以及 (dx_2, dy_2) (單位為單位格 / 秒)，為了兼顧模擬的方便性以及數值的精確性，你決定限制代表速度的四項數值 dx_1, dx_2, dy_1, dy_2 皆為 1 或是 -1 。

暗粒子屬於一種玻色子，所以暗粒子的位置可以與牆壁或是另一顆暗粒子的位置重合（也就是說其座標可以是 $(0, 0)$ ），然而此時將會反彈，反彈的方式遵照一般物體的二維彈性碰撞公式。暗粒子最特別的地方就在於，他的移動不是連續的，也就是說一顆位在 $(2, 4)$ ，速度為 $(1, 1)$ 的暗粒子會在一秒鐘後瞬移到 $(3, 5)$ ，也就是說他和另一顆位在 $(3, 5)$ ，速度為 $(-1, -1)$ 的暗粒子不會發生碰撞。

請找出模擬 t 秒之後，兩顆暗粒子的位置。

註：兩顆一樣的球發生彈性碰撞時會交換速度。

Input Format

輸入有四行：

第一行有兩個整數 N, M ，代表模擬場地在 x 方向與 y 方向的長度。

第二行有四個整數 x_1, y_1, x_2, y_2 代表兩顆粒子的初始位置，保證兩粒子的初始位置不相同。

第三行有四個整數 dx_1, dy_1, dx_2, dy_2 代表兩顆粒子的初始速度。

第四行有一個正整數 t 代表模擬的時間。

- $1 \leq N, M \leq 300$
- $0 \leq x_1, x_2 \leq N$
- $0 \leq y_1, y_2 \leq M$
- $dx_1, dy_1, dx_2, dy_2 \in \{1, -1\}$
- $1 \leq t \leq 10000$

Output Format

請輸出四個整數 x_1, y_1, x_2, y_2 代表兩顆粒子的最終位置，每個數字之間以空白分隔。

Sample Input 1

[copy](#)

```
20 20
3 4 3 6
-1 1 -1 -1
2
```

Sample Input 2

[copy](#)

```
1 10
0 0 1 4
-1 -1 1 1
5
```

Sample Output 1

[copy](#)

```
1 4 1 6
```

Sample Output 2

[copy](#)

```
1 5 0 9
```

Problem	Total Time (ms)	Max Memory (RSS, Kib)	Verdict	Score		
536. 碰粒子	104	3456	Accepted	100		
Subtask Results						
Subtask no.	Testdata Range	Constraints		Score		
1	0-1	範例測試		0/0		
2	0-23	無額外限制		100/100		
Testdata Results						
Testdata no.	Subtasks	Time (ms)	Memory (VSS, Kib)	Memory (RSS, Kib)	Verdict	Score
0	0 0	2.4	6352	3456	Accepted	100
1	0 0	4.1	6352	3456	Accepted	100
2	0	4.1	6352	3456	Accepted	100
3	0	4.6	6352	3456	Accepted	100
4	0	4.4	6352	3456	Accepted	100
5	0	4.4	6352	3456	Accepted	100
6	0	5.0	6352	3456	Accepted	100
7	0	4.2	6352	3456	Accepted	100
8	0	4.9	6352	3456	Accepted	100
9	0	4.3	6352	3456	Accepted	100
10	0	4.3	6352	3456	Accepted	100
11	0	4.1	6352	3456	Accepted	100
12	0	4.2	6352	3456	Accepted	100
13	0	4.5	6352	3456	Accepted	100
14	0	4.1	6352	3456	Accepted	100
15	0	5.1	6352	3456	Accepted	100
16	0	4.5	6352	3456	Accepted	100
17	0	4.0	6352	3456	Accepted	100
18	0	3.8	6352	3456	Accepted	100
19	0	4.4	6352	3456	Accepted	100
20	0	4.0	6352	3456	Accepted	100
21	0	4.5	6352	3456	Accepted	100
22	0	5.0	6352	3456	Accepted	100
23	0	5.1	6352	3456	Accepted	100

Submissions Problem #536

Submit to Problem									
Quick Filter		Enter Username	Enter Problem ID	Select Result	Select Compiler	Filter			
#	PID	Submitter	Time	Memory	Verdict	Compiler	Code Length	Score	Submit Time
67731	536	joseph	104	3456	AC	c++17	1.03 KB	100	2025-11-02 14:48:57
67729	536	joseph	101	3456	WA	c++17	1.03 KB	0	2025-11-02 14:47:04
67668	536	joseph	113	3456	WA	c++17	4.81 KB	0	2025-11-01 17:21:14
67634	536	joseph	93	3456	WA	c++17	313 Bytes	0	2025-10-31 21:00:32

```
int px1 = x1 + dx1;
```

```
int py1 = y1 + dy1;
```

```
int px2 = x2 + dx2;
```

```
int py2 = y2 + dy2; //預計下一步的位置
```

```
if (px1 < 0 || px1 > N){
```

```
    dx1 = - dx1;
```

```
    px1 = x1 + dx1;
```

```
}
```

```
if (py1 < 0 || py1 > M){
```

```
    dy1 = - dy1;
```

```
    py1 = y1 + dy1;
```

```
} //判斷點 1 是否撞牆，如果超出邊界就令速度反向並再計算新位置，點 2
```

重複

```
if (px1 == px2 && py1 == py2){  
    swap(dx1, dx2);  
    swap(dy1, dy2);  
}  
    //若兩個點撞在同一格則交換速度  
  
x1 = px1;  
y1 = py1;  
x2 = px2;  
  
y2 = py2; //套用新的位置
```