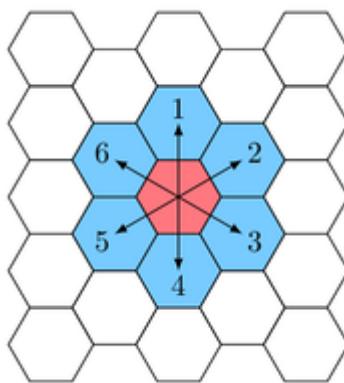


## 六角疆域 (Hexagonal Territory)

在一個六角形磚塊鋪排而成的無窮延伸的鋪磚中，Pak Dengklek 正站在一個六角形磚塊上，稱為初始磚塊。在此鋪磚中，若兩個六角形磚塊共用同一邊，則稱此二個六角形磚塊相鄰。每一步 (step) Pak Dengklek 可以從某個六角形磚塊朝向六個可能的方向之一移動至某一個相鄰磚塊，六個方向由 1 至 6 編號，如下圖所示。



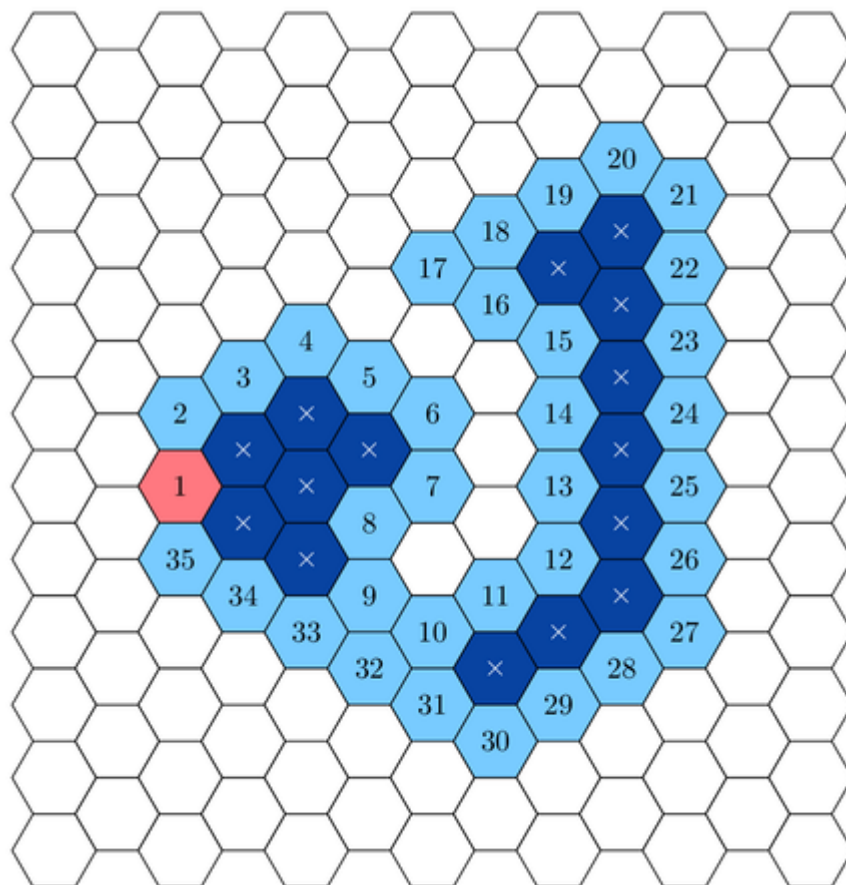
Pak Dengklek 將順著一條路徑來構築一個疆域，此路徑由  $N$  個動作 (move) 拜訪到的一序列六角磚塊構成。第  $i$  個動作為選擇一個方向  $D[i]$  後，往該方向前進  $L[i]$  步。

這個路徑具有下列特性：

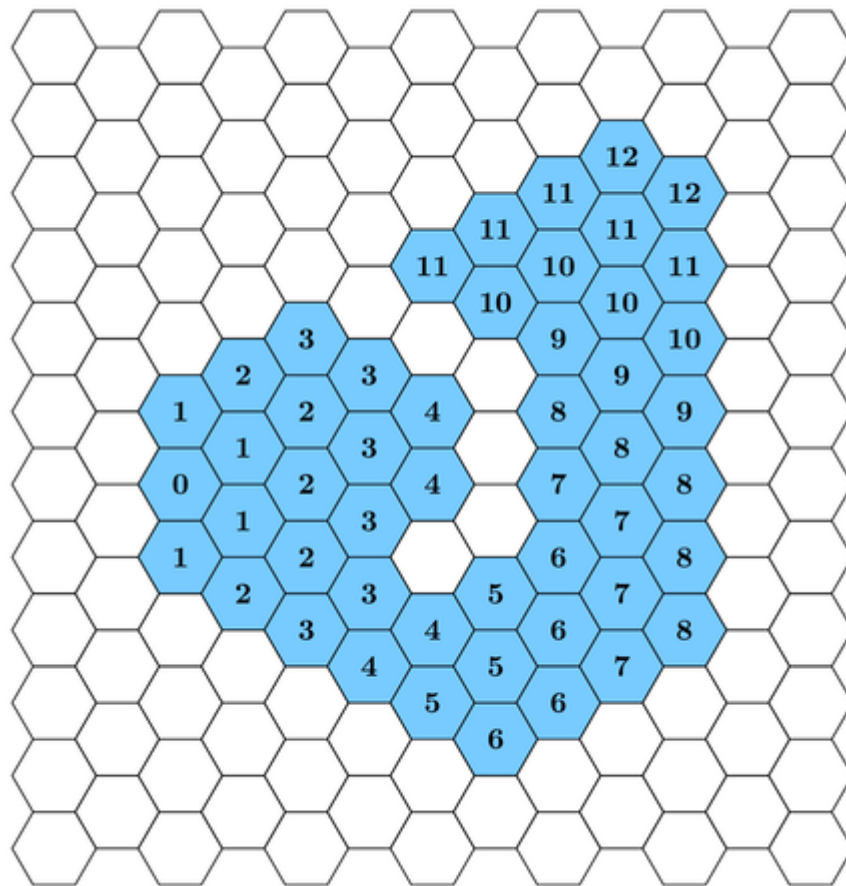
- 此路徑為 *封閉的 (closed)*，表示路徑上最後一個磚塊和路徑上第一個磚塊相同。
- 此路徑為 *簡單的 (simple)*，表示除了初始磚塊外，每個磚塊至多被拜訪一次，初始磚塊恰好被拜訪兩次 (一次在路徑的起點，一次在路徑的終點)。
- 此路徑為 *無遮蔽的 (exposed)*，表示此路徑上每個磚塊都和至少一個不在此路徑上且非 *內部的 (inside)* 的磚塊相鄰。
  - 稱一個磚塊為 *內部的* 若且唯若該磚塊不在此路徑上且從該磚塊出發，在不拜訪到路徑上的任一點的前提下，任意多步都僅能拜訪到有限多個磚塊。

下面是一個 Pak Deklek 可走的路徑範例。

- 編號 1 的磚塊 (粉紅色) 為初始 (且為路徑上最後的) 磚塊。
- 有編號的磚塊 (粉紅色與淺藍色) 為路徑上的磚塊，編號順序為磚塊被拜訪的順序。
- 打叉的磚塊 (深藍色) 為內部的磚塊。



構築完成的疆域由路徑上的磚塊以及內部的磚塊構成。疆域內某個磚塊  $c$  的距離 (distance) 為由初始磚塊經由疆域內的磚塊移動至  $c$  最少所需的步數 (minimu number of steps)。疆域內一個磚塊的分數 (score) 定義為  $A + d \times B$ ，其中  $A$  與  $B$  為 Pak Dengklek 事先決定的常數， $d$  為該磚塊在此疆域內的距離。下面的圖例中顯示了疆域中每個磚塊的距離，此疆域由前述圖例之路徑構築而成。



請幫助 Pak Denglek 計算經由他做的  $N$  個動作後構築而成的疆域中所有磚塊的分數總和。因為分數總和可能相當大，計算其除以  $10^9 + 7$  的餘數即可。

## 實作細節 (Implementation Details)

你應實作下列函式。

```
int draw_territory(int N, int A, int B, int[] D, int[] L)
```

- $N$ : 動作個數。
- $A, B$ : 計算分數所需的常數。
- $D$ : 一長度為  $N$  的陣列，其中  $D[i]$  為第  $i$  個動作的前進方向。
- $L$ : 一長度為  $N$  的陣列，其中  $L[i]$  為第  $i$  個動作前進的步數。
- 此函式應回傳所構築出的疆域之總分。
- 此函式恰被呼叫一次。

## 範例 (Examples)

考慮下列呼叫：

```
draw_territory(17, 2, 3,
               [1, 2, 3, 4, 5, 4, 3, 2, 1, 6, 2, 3, 4, 5, 6, 6, 1],
               [1, 2, 2, 1, 1, 1, 1, 2, 3, 2, 3, 1, 6, 3, 3, 2, 1])
```

這些動作與前述圖例中的動作相同。下表對於此疆域中每個可能的距離值，列出該距離單個磚塊的分數以及該距離所有磚塊的總分。

Distance	Number of cells	Score of each cell	Total score
0	1	$2 + 0 \times 3 = 2$	$1 \times 2 = 2$
1	4	$2 + 1 \times 3 = 5$	$4 \times 5 = 20$
2	5	$2 + 2 \times 3 = 8$	$5 \times 8 = 40$
3	6	$2 + 3 \times 3 = 11$	$6 \times 11 = 66$
4	4	$2 + 4 \times 3 = 14$	$4 \times 14 = 56$
5	3	$2 + 5 \times 3 = 17$	$3 \times 17 = 51$
6	4	$2 + 6 \times 3 = 20$	$4 \times 20 = 80$
7	4	$2 + 7 \times 3 = 23$	$4 \times 23 = 92$
8	5	$2 + 8 \times 3 = 26$	$5 \times 26 = 130$
9	3	$2 + 9 \times 3 = 29$	$3 \times 29 = 87$
10	4	$2 + 10 \times 3 = 32$	$4 \times 32 = 128$
11	5	$2 + 11 \times 3 = 35$	$5 \times 35 = 175$
12	2	$2 + 12 \times 3 = 38$	$2 \times 38 = 76$

所求的總分為  $2 + 20 + 40 + 66 + 56 + 51 + 80 + 92 + 130 + 87 + 128 + 175 + 76 = 1003$ 。因此，函式 `draw_territory` 應回傳 1003。

## 條件限制 (Constraints)

- $3 \leq N \leq 200\,000$
- $0 \leq A, B \leq 10^9$
- $1 \leq D[i] \leq 6$  (for all  $0 \leq i \leq N - 1$ )
- $1 \leq L[i]$  (for all  $0 \leq i \leq N - 1$ )
- $L$  中的元素總和不超過  $10^9$ 。
- 此路徑必為封閉的、簡單的、無遮蔽的。

## 子任務 (Subtasks)

1. (3 points)  $N = 3, B = 0$
2. (6 points)  $N = 3$

3. (11 points)  $L$  中的元素總和不超過 2000。
4. (12 points)  $B = 0$ ,  $L$  中的元素總和不超過 200 000。
5. (15 points)  $B = 0$
6. (19 points)  $L$  中的元素總和不超過 200 000
7. (18 points)  $L[i] = L[i + 1]$  (for all  $0 \leq i \leq N - 2$ )
8. (16 points) 無額外限制。

## 範例評分程式 (Sample Grader)

範例評分程式以下面的格式讀取您的輸入：

- line 1:  $N \ A \ B$
- line  $2 + i$  ( $0 \leq i \leq N - 1$ ):  $D[i] \ L[i]$

範例評分程式以下面的格式輸出您的答案：

- line 1: return value of `draw_territory`