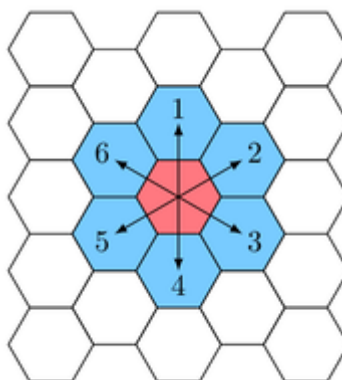


六角形の領域 (Hexagonal Territory)

六角形のマスが敷き詰められた無限に広い平面上に Pak Dengklek がいる。ここで、Pak Dengklek が最初にいるマスのことを「始点のマス」と呼ぶ。また、2つのマスが「隣接している」とは、それらが共通の辺を持つことをいう。1ステップでは、Pak Dengklek は以下の図で1から6までの番号が付けられている方向のうちどれか1つの方向を選び、その方向に隣接するマスに移動する。

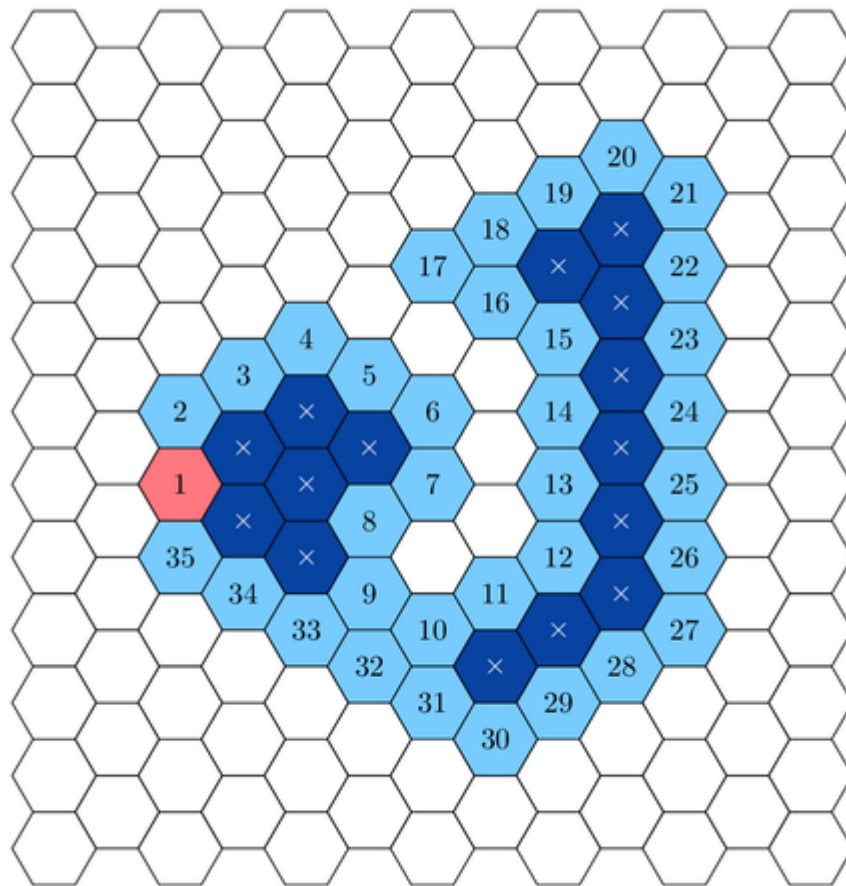


Pak Dengklek は N 回の行動によって訪れたマスから成るパスによって領域を作る。 i 回目の行動では、方向 $D[i]$ に隣接するマスに移動することを $L[i]$ ステップ繰り返す。パスは以下の性質を満たす。

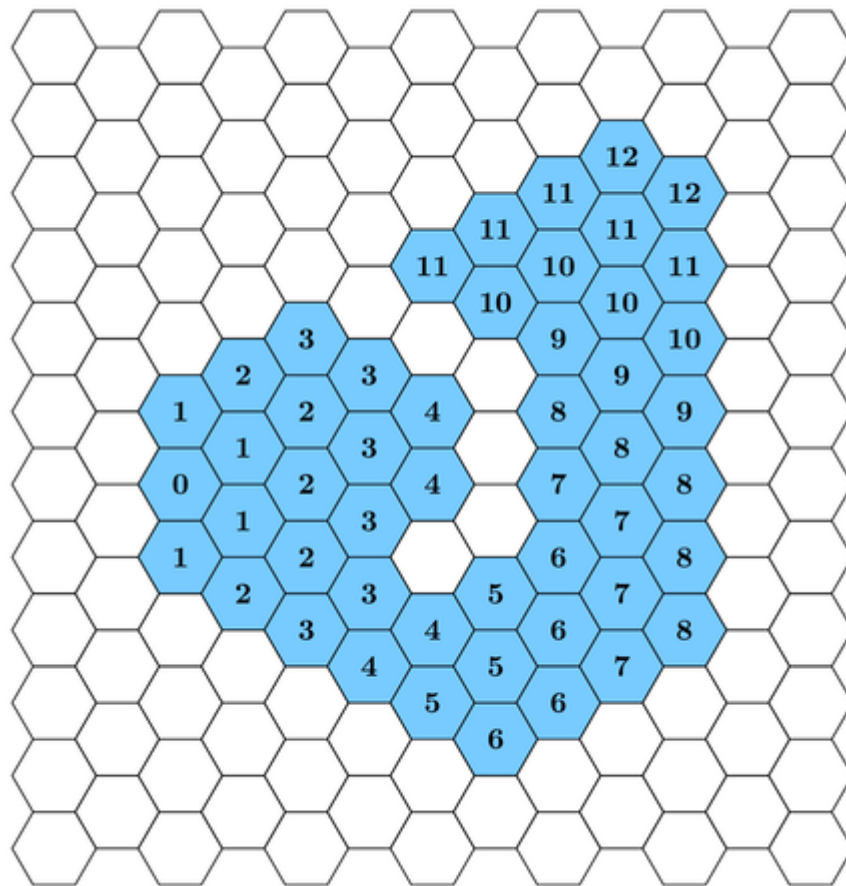
- パスは 閉じている (*closed*)。これは、パスの始点のマスと終点のマスが同じということを表す。
- パスは 単純である (*simple*)。これは、パスがちょうど2回（最初と最後に）訪れる始点のマスを除いて、2回以上訪れるマスが存在しないということを表す。
- パスは 露出している (*exposed*)。これは、パス上の各マスが、パス上でも 内部 (*inside*) でもないマスと隣接していることを表す。
 - マスが 内部 (*inside*) であるとは、そのマスから、パス上にない隣接しているマスへの移動を繰り返したとき、有限個のマスにしか到達できないということを表す。

以下は、Pak Dengklek が移動することのできるパスの例である。

- 1 と書かれたマス（ピンク色）は始点の（かつ終点の）マスである。
- 番号が書かれたマス（水色）はパス上のマスであり、書かれた番号は訪れた順番である。
- バツ印が書かれたマス（紺色）は内部 (*inside*) である。



作られた領域はパス上のマスと内部 (inside) であるマスから構成される．ここで，領域内のマス c の「距離」とは，始点のマスからマス c まで領域内のマスのみを移動するときの最短距離のことである．領域内の各マスの得点は， $A + d \times B$ と定義される． A と B は事前に Pak Dengklek によって定められた定数であり， d はそのマスの距離である．以下の図は，上記の例によって作られた領域内の各マスの距離を表す．



Pak Dengklek の代わりに、 N 回の行動によって彼が作った領域の各マスの得点の合計を求めよ。ただし、答えは十分大きくなる可能性があるため、 $10^9 + 7$ で割った余りを求めよ。

実装の詳細 (Implementation Details)

以下の関数を実装せよ。

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

- N : 行動の回数。
- A, B : 得点計算に用いる定数。
- D : 長さ N の配列であり、 $D[i]$ は i 回目の行動の方向を表す。
- L : 長さ N の配列であり、 $L[i]$ は i 回目の行動で何ステップ移動するかを表す。
- この関数は領域内の各マスの得点の合計を $10^9 + 7$ で割った余りを返さなければならない。
- この関数はちょうど 1 回だけ呼び出される。

例 (Examples)

以下の関数呼び出しを考える。

```
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])
```

この例は上の説明で示したものと同一である。領域内の各マスの得点は距離別に計算すると以下の表のようになる。

距離	マスの個数	各マスの得点	合計得点
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	$2 + 12 \times 3 = 38$	$2 \times 38 = 76$

得点の合計は $2 + 20 + 40 + 66 + 56 + 51 + 80 + 92 + 130 + 87 + 128 + 175 + 76 = 1003$ である。したがって、関数 `draw_territory` は 1003 を返さなければならない。

制約 (Constraints)

- $3 \leq N \leq 200\,000$.
- $0 \leq A, B \leq 10^9$.
- $1 \leq D[i] \leq 6$ ($0 \leq i \leq N - 1$).
- $1 \leq L[i]$ ($0 \leq i \leq N - 1$).
- L の各値の合計は 10^9 以下。
- パスは閉じている (closed)。パスは単純である (simple)。パスは露出している (exposed)。

小課題 (Subtasks)

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

3. (11 点) L の各値の合計は 2000 以下.
4. (12 点) $B = 0$, L の各値の合計は 200 000 以下.
5. (15 点) $B = 0$.
6. (19 点) L の各値の合計は 200 000 以下.
7. (18 点) $L[i] = L[i + 1]$ ($0 \leq i \leq N - 2$).
8. (16 点) 追加の制約はない.

採点プログラムのサンプル (Sample Grader)

採点プログラムのサンプルは以下の形式で入力を読み込む.

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

採点プログラムのサンプルはあなたの解答を以下の形式で出力する.

- 1 行目: `draw_territory` の戻り値.