Permainan Ruang Bawah Tanah

Robert sedang mendesain sebuah permainan komputer baru. Permainan tersebut melibatkan seorang pahlawan, n orang lawan dan $n+1$ ruang bawah tanah atau yang biasa lebih dikenal sebagai *dungeon*. Para lawan dinomori dari 0 sampai $n-1$ dan *dungeon* dinomori dari 0 sampai n. Lawan ke-i ($0 \leq i \leq n-1$) berada di *dungeon* i dan memiliki kekuatan $s[i]$. Tidak terdapat lawan pada *dungeon* n.

Sang pahlawan memulai permainan dengan memasuki *dungeon* x dan memiliki kekuatan awal z. Setiap kali ia memasuki *dungeon* apapun, asumsikan bernomor i ($0 \leq i \leq n-1$), ia akan mengkonfrontasi lawan ke-i, dan salah satu dari dua kemungkinan berikut akan terjadi:

- Jika kekuatan sang pahlawan lebih besar dari atau sama dengan kekuatan lawannya yang bernilai $s[i]$, sang pahlawan menang. Hal ini akan membuat kekuatan sang pahlawan **bertambah** sebanyak $s[i]$ ($s[i] \geq 1$). Dalam kasus ini, sang pahlawan kemudian akan memasuki *dungeon* $w[i]$ ($w[i] > i$).
- Jika tidak, sang pahlawan kalah. Hal ini akan mengakibatkan kekuatan sang pahlawan **bertambah** sebanyak $p[i]$ ($p[i] \geq 1$). Dalam kasus ini, sang pahlawan kemudian akan memasuki *dungeon* $l[i]$.

Perhatikan bahwa $p[i]$ bisa bernilai lebih kecil, sama dengan, atau lebih besar dari $s[i]$. Selain itu, $l[i]$ juga bisa bernilai lebih kecil, sama dengan, atau lebih besar dari i. Apapun hasil dari konfrontasi tersebut, sang lawan akan tetap berada di *dungeon* i dan mempertahankan kekuatan $s[i]$.

Permainan berakhir ketika sang pahlawan memasuki *dungeon* n. Dapat dibuktikan bahwa permainan ini akan berakhir setelah melalui jumlah konfrontasi yang *finite*, terlepas dari nilai kekuatan awal dan *dungeon* tempat sang pahlawan memulai permainan ini.

Robert meminta anda untuk mengetes permainannya dengan menjalankan q buah simulasi. Untuk setiap simulasi, Robert mendefinisikan sebuah *dungeon* awal x and kekuatan awal z. Tugas Anda adalah mencari tahu, untuk setiap simulasi, kekuatan sang pahlawan ketika permainan tersebut berakhir.

Detail Implementasi

Anda perlu mengimplementasikan fungsi berikut:

```
void init(int n, int[] s, int[] p, int[] w, int[] l)
```

- n: jumlah lawan.
- s, p, w, l: *array* yang masing-masing memiliki panjang n. Untuk setiap $0 \leq i \leq n-1$:

    - $s[i]$ adalah kekuatan lawan ke-i. Nilai ini juga merupakan kekuatan yang didapatkan oleh sang pahlawan ketika memenangkan konfrontasi terhadap lawan ke-i.
    - $p[i]$ adalah kekuatan yang didapatkan oleh sang pahlawan ketika kalah dalam konfrontasi terhadap lawan ke-i.
    - $w[i]$ adalah *dungeon* yang dimasuki oleh sang pahlawan setelah menang terhadap lawan ke-i.
    - $l[i]$ adalah *dungeon* yang dimasuki oleh sang pahlawan setelah kalah terhadap lawan ke-i.
- Fungsi ini dipanggil tepat sekali, sebelum pemanggilan apapun terhadap fungsi `simulate` (perhatikan penjelasan dibawah).

```
int64 simulate(int x, int z)
```

- x: *dungeon* yang pertama kali dimasuki oleh sang pahlawan.
- z: kekuatan awal sang pahlawan.
- Fungsi ini harus mengembalikan kekuatan sang pahlawan ketika permainan berakhir, dengan asumsi bahwa ia memulai permainan dengan memasuki *dungeon* x, dan memiliki kekuatan awal z.
- Fungsi ini dipanggil tepat sebanyak q kali.

Contoh

Perhatikan pemanggilan berikut:

```
init(3, [2, 6, 9], [3, 1, 2], [2, 2, 3], [1, 0, 1])
```

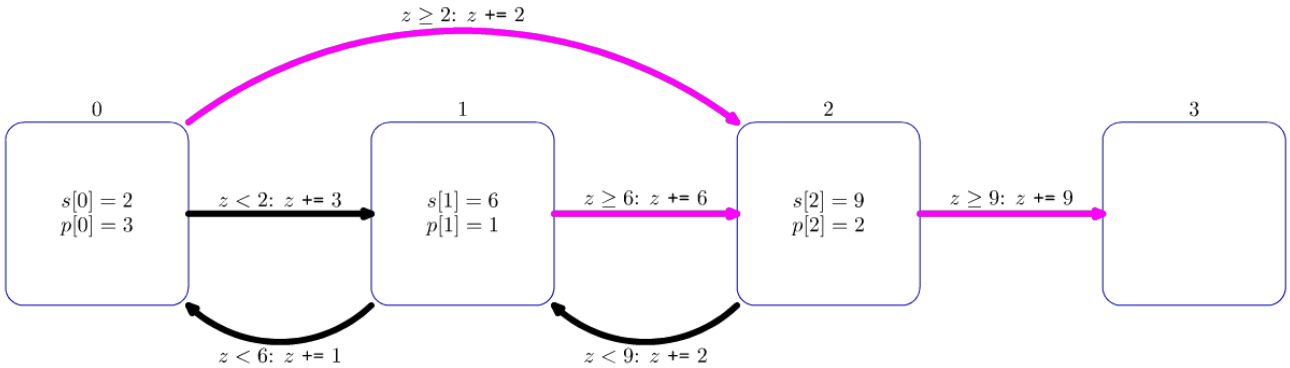

Diagram di atas mengilustrasikan pemanggilan fungsi ini. Setiap persegi melambangkan sebuah *dungeon*. Untuk *dungeon* bernomor 0, 1 and 2, nilai dari $s[i]$ and $p[i]$ diindikasikan di dalam persegi-persegi yang ada. Panah-panah berwarna *magenta* menunjukkan kemana sang pahlawan bergerak setelah memenangkan konfrontasi, sedangkan panah-panah berwarna hitam menunjukkan kemana sang pahlawan bergerak saat kalah.

Misalkan *grader* memanggil `simulate(0, 1)`.

Permainan akan berlangsung sebagai berikut:

Dungeon	Kekuatan sang pahlawan sebelum konfrontasi	Hasil
0	1	Kalah
1	4	Kalah
0	5	Menang
2	7	Kalah
1	9	Menang
2	15	Menang
3	24	Permainan berakhir

Karena itu, pemanggilan fungsi ini harus mengembalikan nilai 24.

Misalkan *grader* memanggil `simulate(2, 3)`.

Permainan akan berlangsung sebagai berikut:

Dungeon	Kekuatan sang pahlawan sebelum konfrontasi	Hasil
2	3	Kalah
1	5	Kalah
0	6	Menang
2	8	Kalah
1	10	Menang
2	16	Menang
3	25	Permainan berakhir

Karena itu, pemanggilan fungsi ini harus mengembalikan nilai 25.

Batasan

- $1 \le n \le 400\,000$
- $1 \le q \le 50\,000$
- $1 \le s[i], p[i] \le 10^7$ (untuk setiap $0 \le i \le n-1$)
- $0 \le l[i], w[i] \le n$ (untuk setiap $0 \le i \le n-1$)
- $w[i] > i$ (untuk setiap $0 \le i \le n-1$)
- $0 \le x \le n-1$
- $1 \le z \le 10^7$

Subsoal

1. (11 poin) $n \le 50\,000,\ q \le 100,\ s[i], p[i] \le 10\,000$ (untuk setiap $0 \le i \le n-1$)

2. (26 poin) $s[i] = p[i]$ (untuk setiap $0 \le i \le n-1$)
3. (13 poin) $n \le 50\ 000$, seluruh lawan memiliki kekuatan yang sama, dengan kata lain, $s[i] = s[j]$ untuk setiap $0 \le i, j \le n-1$.
4. (12 poin) $n \le 50\ 000$, terdapat paling banyak 5 buah bilangan bulat berbeda di antara seluruh nilai $s[i]$.
5. (27 poin) $n \le 50\ 000$
6. (11 poin) Tidak atas batasan tambahan.

Contoh *grader*

Contoh *grader* membaca masukan dengan format berikut:

- baris 1: n q
- baris 2: $s[0]$ $s[1]$ $\ldots$ $s[n-1]$
- baris 3: $p[0]$ $p[1]$ $\ldots$ $p[n-1]$
- baris 4: $w[0]$ $w[1]$ $\ldots$ $w[n-1]$
- baris 5: $l[0]$ $l[1]$ $\ldots$ $l[n-1]$
- baris $6+i$ ($0 \le i \le q-1$): x z untuk pemanggilan ke-i terhadap fungsi `simulate`.

Contoh *grader* mencetak jawaban Anda dengan format berikut:

- baris $1+i$ ($0 \le i \le q-1$) : nilai yang dikembalikan oleh pemanggilan ke-i terhadap fungsi `simulate`.