

## Mario y TuTubo

Contribución de Facundo Martín Gutiérrez

### Descripción del problema

Mario tuvo un día agotador. Luego de reparar incansablemente las tuberías del barrio de Droomklyn durante todo el día, decidió recompensarse con un abundante plato de espagueti. Terminando su día, durante un pequeño momento de ocio, optó por mirar videos de carreras de Kart (uno de sus pasatiempos favoritos) en el famoso sitio “TuTubo”.

En “TuTubo”, al terminar de reproducirse un cierto video se reproduce automáticamente otro video (no necesariamente distinto), cuya sugerencia solo depende del video que estaba reproduciéndose anteriormente.

Nuestro protagonista se quedó dormido durante la quinta vuelta de la sempiterna carrera de la pista Arco Iris. Mientras duerme, queremos poder calcular qué video estará reproduciéndose en el monitor de Mario al cabo de  $k$  videos. Más aún, dado que esta conducta de Mario suele repetirse, queremos poder calcularlo sin importar el video en el que es alcanzado por el sueño.

Más precisamente, tenemos  $N$  videos, numerados desde 0 hasta  $N - 1$  inclusive, y para cada uno de esos videos (digamos el video  $X$ ), conocemos cuál es el próximo que “TuTubo” reproduce si anteriormente se estaba reproduciendo ese video  $X$ . A continuación, tu tarea consiste en responder  $Q$  preguntas de la forma: “¿Cuál es el video que estará reproduciéndose si Mario se durmió viendo el video  $x$  y pasaron  $k$  videos hasta que se despertó?”

### Detalles de implementación

Debes implementar la función `tutubo(siguiete, preguntaX, preguntaK)`:

- `siguiete`: Arreglo de  $N$  enteros entre 0 y  $N - 1$  inclusive, que para cada video  $i$  indica en `siguiete[i]` cuál es el que se reproduce a continuación del  $i$ .
- `preguntaX`, `preguntaK`: Arreglos de  $Q$  enteros. Para cada  $i$  entre 0 y  $Q - 1$  inclusive, la pregunta  $i$ -ésima (contando desde 0) se debe responder con el video que estará reproduciéndose si pasan `preguntaK[i]` videos desde el video `preguntaX[i]`.

Debe retornar un arreglo de  $Q$  enteros, con las respuestas a las preguntas en el mismo orden en que se dan en los parámetros de entrada.

### Cotas

- $1 \leq N \leq 100.000$
- $1 \leq Q \leq 100.000$
- $0 \leq \text{preguntaX}[i] < N$
- $1 \leq \text{preguntaK}[i] \leq 10^{18}$

### Evaluable local

El evaluador local lee de la entrada estándar:

- Dos enteros  $N$  y  $Q$
- $N$  líneas, cada una con un entero `siguiete[i]`
- $Q$  líneas más, cada una con dos enteros `preguntaX[i]` y `preguntaK[i]`

Escribe a la salida estándar el resultado retornado por la función, con un elemento por línea.

**Ejemplos**

Si el evaluador local recibe la siguiente entrada:

4	4
1	
2	
0	
1	
0	3
0	4
3	2
3	10

Para una implementación correcta escribirá:

0
1
2
1

**Subtareas**

1.  $Q = 1$  y además  $preguntaK[i] = 1$  para todo  $i$  (2 ptos.)
2.  $preguntaK[i] = 1$  para todo  $i$  (3 ptos.)
3.  $Q = 1$  (11 ptos.)
4.  $preguntaK[i] \leq 50$  para todo  $i$  (15 ptos.)
5.  $N, Q \leq 1.000$  (24 ptos.)
6.  $N, Q \leq 15.000$  y  $preguntaK[i] \leq 10^6$  para todo  $i$  (20 ptos.)
7. Sin más restricción (25 ptos.)