

# Torre de Cubos

(Basado en un problema de la ACM Programming Contest)

## 1. Descripción del problema

Se tienen  $N$  cubos de colores de masas distintas. Cada una de las caras de los cubos posee un color. El objetivo es construir la torre de cubo más alta posible, sujeta a las siguientes restricciones:

- No se puede colocar un cubo más pesado sobre uno más ligero.
- La cara de abajo de cada cubo (excepto para el primer cubo de la torre) debe tener el mismo color que la cara de arriba del cubo que se encuentra debajo de este.

Para solucionar este problema debe crear un programa llamado `TorreDeCubos.java` que se debe poder ejecutar desde la consola con el siguiente comando:

```
>java TorreDeCubos <instancia>
```

donde `instancia` es el nombre del archivo con la entrada de los datos.

### 1.1. Entrada de los datos

Los datos se ingresarán a su programa por medio de un archivo de texto con las siguientes características. La primera línea posee un entero  $N$  ( $1 \leq N \leq 500$ ), que indica el número total de cubos. Las siguientes  $N$  líneas contienen la descripción de cada cubo. Un cubo es caracterizado con los colores de sus caras, los cuales son dados en el siguiente orden: *frente*, *detrás*, *izquierda*, *derecha*, *arriba* y *abajo*. Los colores son identificados por números enteros en un rango de 1 a 100. Se debe asumir que los cubos aparecen en el archivo en orden ascendiente a sus masas, es decir, el cubo 1 es el más liviano y el  $N$  es el más pesado.

### 1.2. Salida de los datos

La primera línea corresponde al número de cubos que forman parte de la torre más alta posible. Las siguientes líneas describen a los cubos que componen la torre desde arriba hacia abajo, con una descripción por línea. Cada descripción esta compuesta en primer lugar con el número del cubo en el archivo de entrada, seguido por un espacio en blanco y luego un *string* que indica cual es la cara del cubo que se encuentra arriba en la torre. Los *strings* que corresponden a las caras del cubo son: *frente*, *detrás*, *izquierda*, *derecha*, *arriba* o *abajo*. Pueden haber múltiples soluciones, cualquiera de ellas es aceptable.

### 1.3. Ejemplos prácticos

#### 1.3.1. Ejemplo 1

Sea un archivo llamado `cubos1.txt`, con el siguiente contenido:

```
3
1 2 2 2 1 2
3 3 3 3 3 3
3 2 1 1 1 1
```

Su programa debe mostrar el siguiente resultado por la salida estándar:

```
2
2 frente
3 frente
```

#### 1.3.2. Ejemplo 2

Sea `cubos2.txt` un archivo con el siguiente contenido:

```
10
1 5 10 3 6 5
2 6 7 3 6 9
5 7 3 2 1 9
1 3 3 5 8 10
6 6 2 2 4 4
1 2 3 4 5 6
10 9 8 7 6 5
6 1 2 3 4 7
1 2 3 3 2 1
3 2 1 1 2 3
```

Su programa debe imprimir por la salida estándar el siguiente resultado:

```
8
1 abajo
2 detras
3 derecha
4 izquierda
6 arriba
8 frente
9 frente
10 arriba
```

## 2. Condiciones de entrega del laboratorio

Debe entregar en la página del curso en el aula virtual, el día 16 de Noviembre de 2016 antes de las 4:30 pm, un archivo comprimido llamado **LabSem10-X-Y.tar.gz** con todos los códigos en Java y el archivo Makefile correspondiente. Las letras **X** y **Y** del archivo comprimido son los número de carné de los integrantes del equipo.