

## Cartas

Contribución de Hugo Ryckeboer

### Descripción del problema

Un amigo, fanático de los solitarios, jugó reiteradamente a uno muy simple que consiste en repetir los siguientes dos pasos hasta quedarse con una sola carta:

- Toma la carta de arriba del mazo y la descarta
- Luego **K** veces repite la acción de tomar la carta de arriba del mazo y colocarla abajo.

El valor de **K** es fijado una única vez por toda la duración del solitario. Cuando juega con mazos voluminosos a veces se distrae en la cuenta de **K**. Para controlar su correcta ejecución pide que le armes una función `solitario` que lo resuelva en paralelo con él y le indique el valor de la carta final.

Las cartas con las que juega son variadas, no necesariamente francesas ni españolas con lo cual para facilitar su ingreso cada figura está identificada por un número entero dentro del rango **1** a **10.000**. Le gusta jugarlo con mazos voluminosos, de tamaño **N**, y como no quiere perder tiempo dictándote grandes secuencias de números decide armar sus mazos de la siguiente forma: Mezcla **M** cartas y luego con cartas adicionales reproduce copias idénticas que amontona hasta que las pilas **2M**, **3M**, alcancen para extraer desde abajo las **N** que necesita para el juego. Las **M** cartas las describe de abajo hacia arriba.

Para ayudar al amigo se te pide que envíes en un archivo `solitario.pas` o `solitario.cpp` tu implementación de la función `solitario( c, N, K )` que devuelve un ENTERO y cuyos parámetros son:

- **c** : ARREGLO de **M** ENTEROS en el rango **1..10.000**, no necesariamente distintos.
- **N** : un ENTERO LARGO indicando la cantidad de cartas con las cuales quiere jugar el solitario.
- **K** : un ENTERO con la constante elegida para esa partida.

### Ejemplo

Si los datos fueran **M** = 3 , **N** = 7 , **K** = 1.

16	4	21
----	---	----

Primero construiría un mazo triple:

16, 4, 21, 16, 4, 21, 16, 4, 21  
y de él separaría desde abajo las primeras 7 cartas:

16, 4, 21, 16, 4, 21, 16

tras una jugada le quedaría:

21, 16, 4, 21, 16, 4

tras dos jugadas le quedaría:

16, 21, 16, 4, 21

y siguiendo así el resultado del juego sería: 4

### Cotas

$$1 \leq M \leq 100.000$$

$$0 \leq K \leq 100.000$$

$$1 \leq N \leq 1.000.000.000.000.000.000$$

### Evaluador local

El evaluador local lee la entrada por `stdin` en el siguiente formato:

- Una línea con los valores de **M**, **K** y **N**.
- Una o más líneas conteniendo los **M** valores que componen el mazo.
- Entrega por `stdout` (pantalla) el valor del juego (el valor de la carta final que es una de las **M** descriptas)

### Ejemplo

El archivo `solitario.in` suministrado contiene:

3	1	7
16	4	21

Y con él, la ejecución devuelve:

4
---

### Subtarea

En un número de casos por valor de **45** puntos la cantidad **N** de cartas no superará a **100.000**

De esos casos, un subconjunto por **20** puntos tendrá además un valor de **K** que no supere **100**.

Por otra parte, existirán casos por **30** puntos en los cuales la cantidad **N** de cartas será mayor a **100.000**, pero se tendrá **K** = 1.