Grupo Weboo
Universidad de La Habana

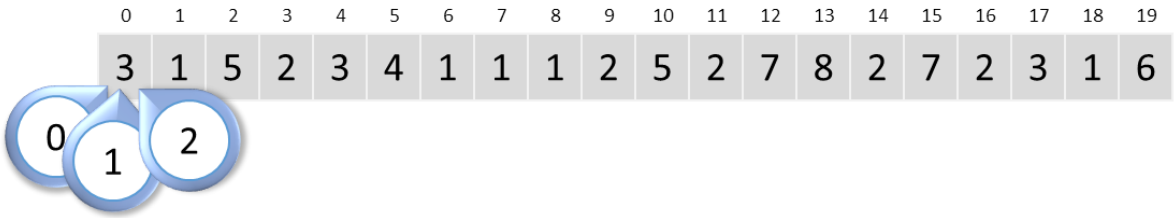
Primer Examen de Programación
Curso 2017-2018

Un parchís un poco raro…

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Sobre un tablero unidimensional de N casillas se ha dispuesto una secuencia de valores enteros. Las posiciones del tablero están enumeradas desde la casilla 0 hasta la casilla N-1. Un total de K jugadores se posicionan en la casilla 0 al principio del juego. Los jugadores, enumerados desde 0 hasta K-1, jugarán por turnos repetitivamente en el orden en que están dispuestos: 0, 1, 2, … K-1, 0, 1, 2 ….

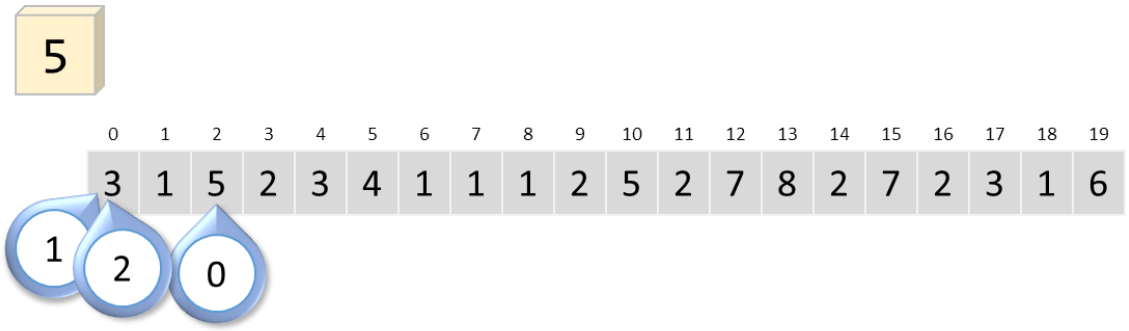
La siguiente figura muestra un ejemplo de tablero, todos los jugadores comienzan en la posición **0**.



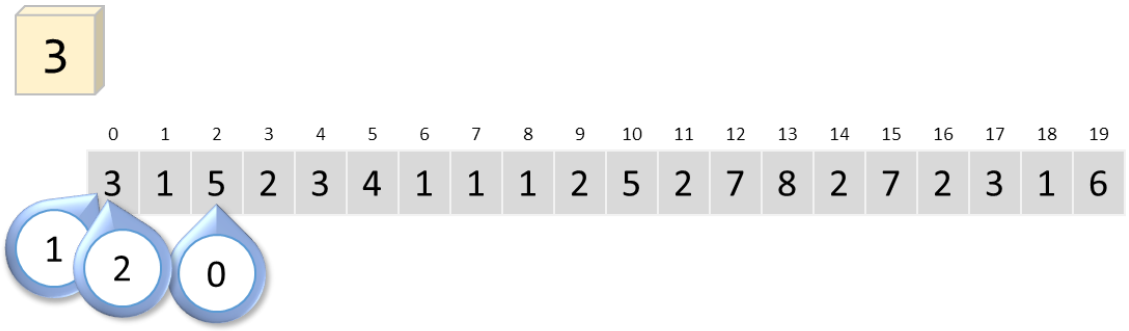
En cada turno el jugador correspondiente lanza un dado especial. A diferencia de un dado tradicional de 6 caras que puede obtener valores entre 1 y 6, este dado permite obtener **cualquier** valor entero. El jugador luego de su lanzamiento deberá moverse a la siguiente posición con valor igual al obtenido en el dado. Si el dado obtuviera el mismo valor de la casilla actual del jugador, éste **no se moverá**. Gana el primer jugador que obtenga un valor que no se encuentre en las casillas restantes **a partir** de su posición.

Por ejemplo, dada la configuración anterior, para las tiradas: 5, 3, 7, 1, 2, 2; la disposición del tablero se modificaría como se muestra a continuación:

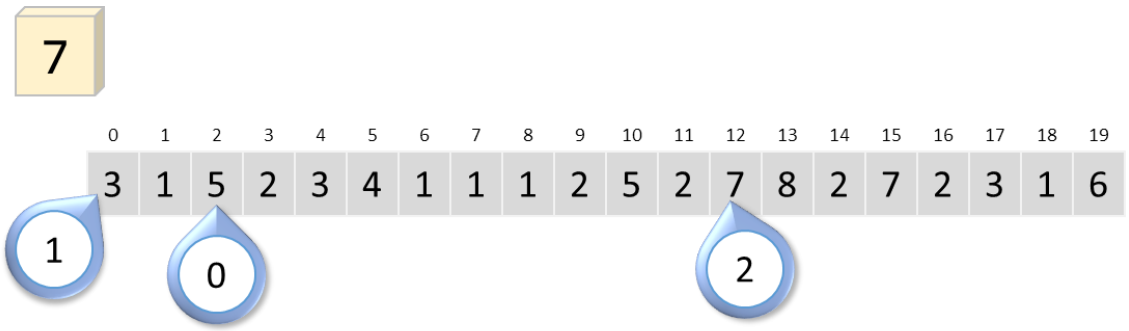
El jugador 0 comienza y obtiene 5, luego se mueve hacia el próximo 5 que se encuentra en la casilla 2.



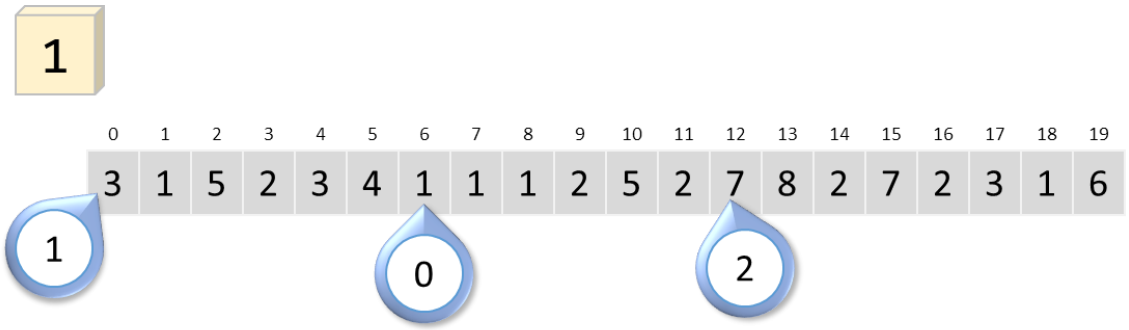
El jugador 1 le sigue y obtiene 3, por lo tanto **no** se mueve puesto que ya está en una casilla con valor 3.



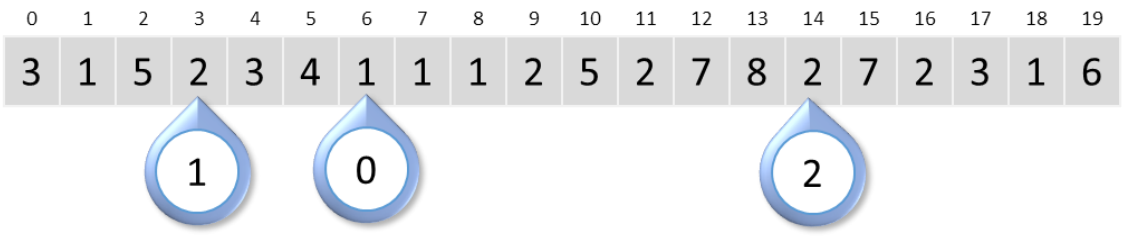
Le sigue el jugador 2 y obtiene 7, moviéndose hasta la posición 12 del tablero.



De vuelta el jugador 0 obtiene 1 y se posiciona en la casilla 6.



Luego el jugador 1 obtiene un 2 para moverse a la posición 3, y por último el jugador 2 obtiene otros 2 para posicionarse en la 14. Luego de esta simulación el tablero queda como sigue:



Luego de estos lanzamientos, si el jugador 0 obtuviera 4, ganaría.

Su tarea consiste en implementar un método que permita simular este juego.

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Usted debe implementar el método `Simula` que se encuentra en la clase `ParchisRaro` en el *namespace* `Weboo.Examen`. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class ParchisRaro
    {
        public static int[] Simula(int[] tablero, int cantidadDeJugadores, int[] lanzamientos)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

Este método tiene como primer parámentro un `int[]` con los valores de las casillas del tablero. Los restantes parámetros son un `int` indicando cuántos jugadores juegan y un `int[]` con la secuencia de lanzamientos del dado que se quiere probar. Su método devolverá un `int[]` con las posiciones de cada jugador (desde el jugador 0 hasta el jugador `cantidadDeJugadores - 1`).

Atención: Si durante la simulación algún jugador gana, entonces su posición será igual al número de casillas del tablero (indicando que el mismo está fuera del tablero) y la simulación **para** (no se contemplan más lanzamientos).

Algunas consideraciones:

- El tablero **siempre** tendrá al menos una casilla y existirá al menos un lanzamiento de dados.
- Existirá **siempre** al menos un jugador.
- Los valores de los lanzamientos **no tienen por qué** estar en el tablero.
- Los valores del tablero y de los lanzamientos serán **siempre** distintos de los valores especiales `int.MinValue` e `int.MaxValue`.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará. En particular, es imprescindible que usted no cambie los parámetros del método `Simula`, ni su orden. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

Ejemplos

```
//EJEMPLO 1
int[] tablero1 = { 3, 1, 5, 2, 3, 4, 1, 1, 1, 2, 5, 2, 7, 8, 2, 7, 2, 3, 1, 6 };
int cantidadDeJugadores1 = 3;
int[] lanzamientos1 = { 5, 3, 7, 1, 2, 2 };
int[] posiciones1 = ParchísRaro.Simula(tablero1, cantidadDeJugadores1, lanzamientos1);
//posiciones1 = { 6, 3, 14 }
//jugador 0 en la posición 6
//jugador 1 en la posición 3
//jugador 2 en la posición 14
//se simularon todos los lanzamientos


//EJEMPLO 2
int[] tablero2 = { 3, 1, 5, 2, 3, 4, 1, 1, 1, 2, 5, 2, 7, 8, 2, 7, 2, 3, 1, 6 };
int cantidadDeJugadores2 = 3;
int[] lanzamientos2 = { 5, 3, 7, 1, 2, 2, 4, 5, 6, 23 };
int[] posiciones2 = ParchísRaro.Simula(tablero2, cantidadDeJugadores2, lanzamientos2);
//posiciones2 = { 20, 3, 14 }
//jugador 0 ganó entonces posición = cantidad de casillas = 20
//jugador 1 en la posición 3
//jugador 2 en la posición 14
//se desechan los últimos 3 lanzamientos ya que gana el jugador 0 antes


//EJEMPLO 3
int[] tablero3 = { 3, 1, 5, 2, 3, 4, 1, 1, 1, 2, 5, 2, 7, 8, 2, 7, 2, 3, 1, 6 };
int cantidadDeJugadores3 = 3;
int[] lanzamientos3 = { 5, 3, 7, 1, 2, 2, 2, 5, 6, 23 };
int[] posiciones3 = ParchísRaro.Simula(tablero3, cantidadDeJugadores3, lanzamientos3);
//posiciones3 = { 20, 10, 19 }
//jugador 0 ganó entonces posición = cantidad de casillas = 20
//jugador 1 en la posición 10
//jugador 2 en la posición 19
//se simularon todos los lanzamientos


//EJEMPLO 4
int[] tablero4 = { 150, -5, 23, 0 };
int cantidadDeJugadores4 = 100;
int[] lanzamientos4 = { -5, 1 };
int[] posiciones4 = ParchísRaro.Simula(tablero4, cantidadDeJugadores4, lanzamientos4);
//posiciones4 = { 3, 4, 0, 0, 0, 0, ..., 0 }
//jugador 0 en la posición 1
//jugador 1 ganó entonces posición = cantidad de casillas = 4
//jugador 2 en la posición 0 (nunca jugó)
//jugador 3 en la posición 0 (nunca jugó)
//jugador k (1<k<100) en la posición 0 (nunca jugó)
//se simularon todos los lanzamientos
```

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.