

Ejercicio práctico a entregar: Tipos Abstractos de Datos

1. Implementá el TAD Queue of T visto en clase(*), utilizando la idea de listas enlazadas con punteros a nodos, de manera tal que **todas las operaciones sean de orden constante**. Para ello, podés representar la cola mediante dos punteros: uno al primer elemento y otro al último.

(*) <https://wiki.cs.famaf.unc.edu.ar/lib/exe/fetch.php?media=algo2:main:06.tipos.abstractos.2.pdf>

2. El Martín Pescador Extreme
(basado en el problema de Josefo)

El juego "Martín Pescador Extreme" consiste en una serie de "rondas" y comienza con una cola de **N** jugadores que inicialmente tendrán **h vidas** cada uno. En cada ronda se re-encola (se manda al fondo) **k veces** a las personas que se encuentren al principio de la cola y, al terminar la ronda, se *le quita una vida* al jugador que haya quedado al principio de la cola y se evalúa lo siguiente sobre este jugador:

1. Si tiene al menos una vida, sigue en el juego y queda en el mismo lugar.
2. Si no tiene más vidas se lo elimina de la cola (pierde)

Al finalizar esta evaluación comienza una nueva ronda.

El juego termina cuando queda una sola persona en la cola, a quien consideramos ganador/a.

Nota: El número **k** es fijo durante todo el juego, es decir, es el mismo en cada ronda.

Ejemplo: Supongamos que la cantidad de vidas inicial de los jugadores es **h = 1**, la cantidad de veces que se re-encola por ronda es **k = 2**, y los jugadores inicialmente son **["Julia", "Matias", "Lucía", "Raul"]**

Ronda 1: (Cola inicial: **["Julia", "Matias", "Lucía", "Raul"]**)

-- re encolado de k personas

["Matias", "Lucía", "Raul", "Julia"] -- se re-encola a "Julia"

["Lucía", "Raul", "Julia", "Matias"] -- se re-encola a "Matias"

-- fin re encolado

-- eliminación de la primera jugadora (Lucía)

["Raul", "Julia", "Matias"]

Ronda 2: (Cola inicial: **["Raul", "Julia", "Matias"]**)

-- re encolado

["Julia", "Matias", "Raul"] -- se re-encola a "Raul"

["Matias", "Raul", "Julia"] -- se re-encola a "Julia"

-- fin re encolado

-- eliminación del primer jugador (Matías)

```
["Raul", "Julia"]
```

Ronda 3: (Cola inicial: ["Raul", "Julia"])

-- re encolado

```
["Julia", "Raul"] -- se re-encola a "Raul"
```

```
["Raul", "Julia"] -- se re-encola a "Julia"
```

-- fin re encolado

-- eliminación del primer jugador (Raúl)

```
["Julia"] --
```

Acá termina el juego y gana "Julia".

Se pide:

Completá la implementación del algoritmo ExtremeKingfisherGame en la cual deberás **utilizar** los TADs Player y Queue of Player **sin romper la abstracción**. Sólo se pueden utilizar operaciones que estén especificadas en los respectivos TADs.

```
fun ExtremeKingfisherGame(playersNames: array[1..N] of String, lives: Nat, k:
Nat) ret winner: String
    var q: Queue of Player
    {- completar -}
end fun
```

La especificación del TAD Player es como sigue:

spec Player where

constructors

```
fun createPlayer(name: String, lives: Nat) ret Player
```

operations

```
fun getName(p: Player) ret n: String
{- Devuelve el nombre del jugador p -}
```

```
fun isAlive(p: Player) ret b: Bool
{- Devuelve True si el jugador p tiene al menos 1 vida -}
```

```
proc loseLife(in/out p: Player)
{- Resta una vida al jugador p -}
{- PRE: getLives(p) > 0 -}
```

end spec

Además de las operaciones anteriores, también se incluyen destroy y copy, como en todos los TADs.