1. Array operations

Head

Implementa una función `head` (inmutable), tal que, dado un array como entrada extraiga y devuelva su primer elemento. Utiliza destructuring.

```
const head = (/* array */) => {}; // Implementation here.
```

Tail

Implementa una función `tail` (inmutable), tal que, dado un array como entrada devuelta todos menos el primer elemento. Utiliza rest operator.

```
const tail = (/* array */) => {}; // Implementation here.
```

Init

Implementa una función `init` (inmutable), tal que, dado un array como entrada devuelva todos los elementos menos el último. Utiliza los métodos que ofrece Array.prototype.

```
const init = (/* array */) => {}; // Implementation here.
```

Last

Implementa una función `last` (inmutable), tal que, dado un array como entrada devuelva el último elemento.

```
const last = (/* array */) => {}; // Implementation here.
```

2. Concat

Implementa una función `concat` (inmutable) tal que, dados 2 arrays como entrada, devuelva la concatenación de ambos. Utiliza rest / spread operators.

```
const concat = (a, b) => {}; // Implementation here.
```

Opcional

Implementa una versión del ejercicio anterior donde se acepten múltiples arrays de entrada (más de 2).

3. Clone Merge

Clone

Implementa una función `clone` que, a partir de un objeto de entrada `source` devuelva un nuevo objeto con las propiedades de `source`:

```
function clone(source) {
  // Implementation here.
}
```

Merge

Implementa una función `merge` que, dados dos objetos de entrada `source` y `target`, devuelva un nuevo objeto con todas las propiedades de `target` y de `source`, y en caso de propiedades con el mismo nombre, `source` sobreescribe a `target`.

Por ejemplo, dados estos 2 objetos:

```
const a = { name: "Maria", surname: "Ibañez", country: "SPA" };
const b = { name: "Luisa", age: 31, married: true };
```

el resultado de mezclar `a` sobre `b` sería:

```
merge(a, b); // {name: "Maria", age: 31, married: true, surname: "Ibañez",
country: "SPA"}
```

TIP: Puedes usar la función "clone" del apartado A.

```
function merge(source, target) {
  // Implementation here.
}
```

4. Read Books

Crea una función isBookRead que reciba una lista de libros y un título y devuelva si se ha leído o no dicho libro. Un libro es un objeto con `title` como string y `isRead` como booleano. En caso de no existir el libro devolver `false` TIP: Existe un método de Array.prototype que te ayudará a buscar según un patrón.

```
function isBookRead(books, titleToSearch) {
  // Implementation here
}
```

Ejemplo

```
const books = [
  { title: "Harry Potter y la piedra filosofal", isRead: true },
  { title: "Canción de hielo y fuego", isRead: false },
  { title: "Devastación", isRead: true },
];

console.log(isBookRead(books, "Devastación")); // true
console.log(isBookRead(books, "Canción de hielo y fuego")); // false
console.log(isBookRead(books, "Los Pilares de la Tierra")); // false
```

Opcional

Utiliza Typescript para añadir los tipos adecuados.

5. Slot Machine

El objetivo de este ejercicio es crear una máquina tragaperras utilizando clases donde cada vez que juguemos insertemos una moneda. Cada máquina tragaperras (instancia) tendrá un contador de monedas que automáticamente se irá incrementando conforme vayamos jugando.

Cuando se llame al método play el número de monedas se debe incrementar de forma automática y debe generar tres booleanos aleatorios que representarán el estado de las 3 ruletas. El usuario habrá ganado en caso de que los tres booleanos sean `true`, y por tanto deberá mostrarse por consola el mensaje:

```
"Congratulations!!!. You won <número de monedas> coins!!";
```

y reiniciar las monedas almacenadas, ya que las hemos conseguido y han salido de la máquina. En caso contrario deberá mostrar otro mensaje:

`"Good luck next time!!".`

Ejemplo de uso

```
class SlothMachine {
  /* ... */
}

const machine1 = new SlothMachine();
machine1.play(); // "Good luck next time!!"
machine1.play(); // "Good luck next time!!"
machine1.play(); // "Congratulations!!!. You won 3 coins!!"
machine1.play(); // "Good luck next time!!"
machine1.play(); // "Congratulations!!!. You won 2 coins!!"
```