

## Escuela de Ingeniería

**Parcial de: Programación I - PRESENCIAL**

**Código de materia:**

**Fecha: Diciembre 2021**

**Grupo: NOCTURNO**

**Hoja 1 de 2**

**V07**

**Duración:**

**2 horas**

**Con material: NO**

**Puntaje máximo:**

**35 puntos**

**1) (6 puntos)** Implementar en JS una función que recibe un array y retorna true si la concatenación de los elementos (exceptuando el último y tomados en el mismo orden en que aparecen) coincide exactamente con el que está en el último lugar, en otro caso retorna false.

**Ejemplos:**

Recibe ["mi", "nombre", "es", "ana", "minombreesana"] retorna true

Recibe ["1", "456", "23", "134456238"] retorna false

**2) (8 puntos)** Implementar en JS una función que recibe dos arrays con números. Se desea verificar que el segundo array contenga al menos todos los valores del primer array o a lo sumo que falte uno. Se asume no hay repetidos. Se pide:

- si falta más de un número, retornar "faltan más"
- si no falta ninguno, retornar "no falta ninguno".
- si falta un solo número, retornar "falta " y el número correspondiente

**Ejemplos:**

Recibe [ 8, 5, 3, 1, 2] y [5, 2, 0, 8, 3, 1, 4, 15], retorna "no falta ninguno"

Recibe [ 5, 2, -5, -2, 28, 0] y [0, 8, 5, 35, -5], retorna "faltan más"

Recibe [5, 2, 3, 1] y [6, 7, 1, 5, 4, 3], retorna "falta 2"

**3)** Se diseñó una página para una competencia de perros. Hay 10 pruebas (numeradas de 1 a 10) y en cada una de ellas, se le asigna un único puntaje al perro. Se tienen definidas estas clases en el archivo clases.js (el código se muestra parcial, se incluye solamente lo necesario para la letra del ejercicio):

<pre>class Perro{   constructor(nombre, edad){     this.nombre = nombre;     this.edad= edad ;   } }</pre>	<pre>class Sistema{   constructor(){     this.listaPerros = [];     this.listaEvaluaciones=[];   }   // pedido 3a   perrosGanadores(){     // codigo a completar   } }</pre>	<pre>class Evaluacion{   constructor(numeroPrueba, perro, puntaje, detalle){     this.numeroPrueba =numeroPrueba;     this.perro= perro ;// objeto perro     this.puntaje=puntaje;     this.detalle = detalle;   } }</pre>
--	--	--

**3a) (máximo 10 puntos)** En la clase Sistema, implementar el método perrosGanadores que retorna un array con los perros que tienen las 10 evaluaciones registradas y su suma de puntajes es máxima. Debe estar ordenado por nombre creciente.

**3b) (máximo 2 puntos)** Asumir disponible la página HTML y el archivo CSS correspondiente. Se tiene este código HTML (parcial):

```
<p id="idParrafo"></p>
<input id="idBoton" type="button" value="Consulta">
<select id="idCombo">
</select>
```

## Escuela de Ingeniería

**Parcial de: Programación I - PRESENCIAL**

**Código de materia:**

**Fecha: Diciembre 2021**

**Grupo: NOCTURNO**

**Hoja 2 de 2**

**V07**

Anotar el CSS a incluir en el archivo de estilo para que solamente ese combo tenga tamaño de letra 15px y todos los párrafos de la página estén en color azul.

**3c) (máximo 9 puntos)** Al presionar el botón se cargará en el combo "idCombo", los datos de los perros ganadores. Para ser ganador, tiene que tener todas las evaluaciones y la suma de todos sus puntajes debe ser el máximo. Asumir disponible y utilizar el método de perrosGanadores de 3a).

De cada perro poner su nombre en mayúsculas y si es "cachorro" (edad menor a 2), "joven" (edad entre 2 y 5, ambos inclusive) o "adulto" (edad mayor a 5). Además, en el párrafo "idParrafo" poner la cantidad de ganadores (ej. "Cantidad de ganadores: 4"). Si no hay datos, en el párrafo poner "sin datos" y dejar el combo vacío.

En el archivo fuentes.js se tiene:

```
window.addEventListener("load", inicio);
let sistema = new Sistema(); // contiene la instancia de Sistema
function inicio(){
    document.getElementById("idBoton").addEventListener("click", resultados);
}
// parte 3c)
function resultados(){
    // código a completar
}
```