

## Escuela de Ingeniería

**Parcial de: Programación I**

**Código de materia:**

**Fecha: Diciembre 2020**

**Grupo: MATUTINO**

**Hoja 1 de 2**

**v11**

**Duración: 2 horas Con material: Si Puntaje máximo: 35 puntos**

INSTRUCCIONES: subir en Aulas UN UNICO ARCHIVO .TXT que incluya el nombre del autor y el código de cada ejercicio.

**1) (14 puntos)** En una tienda de artículos navideños están preparando árboles decorados personalizados. Se tiene en un array el código y precio de cada elemento disponible en la tienda. En otro array, se tiene la descripción de los elementos que se pusieron en un determinado árbol. Los primeros 4 caracteres de la descripción son el código del elemento.

Retornar el precio final del árbol y la indicación de si incluye todos los elementos disponibles o no. El texto a retornar debe tener este formato, siendo XX el costo: "Cuesta XX y tiene todos los elementos" o "Cuesta XX y no tiene todos los elementos".

Ejemplos:

Elementos disponibles: ["GLOB", 100, "PUNT", 250, "GUIA", 200, "SOPO", 50, "CAMP", 80]

Arbol:

["PUNTERO", "GLOBO ROJO", "GLOBO AZUL", "GLOBO ROJO", "GUIA DORADA", "GUIA PLATEADA"]

Retorna: "Cuesta 950 y no tiene todos"

Elementos disponibles: ["GLOB", 100, "PUNT", 30]

Arbol:

["PUNTERO", "GLOBO ROJO", "GLOBO AZUL", "GLOBO ROJO", "GLOBO VERDE", "GLOBO ROJO"]

Retorna: "Cuesta 530 y tiene todos"

**function decoracion (elementos, arbol)**

**2)** Se diseñó una página para la gestión de reservas de un club. En la página se encuentran asociados los archivos clases.js, fuentes.js y estilos.css.

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 Reserva{   constructor(dia, deporte){     // el dia va de 1 a 31     // el deporte de 1 a 20     this.dia = dia;     this.deporte= deporte;   } }</pre>	<pre>class Socio{   constructor (nombre){     this.nombre=nombre;     this.reservas=[];   }   agregarReserva(reserva){     // recibe un objeto de la clase Reserva     this.reservas.push(reserva);   }   // pedido 2a   cantidadReservas(desde, hasta, deporte){     // codigo a completar   } }</pre>	<pre>class Sistema{   constructor(){     this.listaSocios = [];   }   // pedido 2b   sociosConMasDe(cantidad){     // codigo a completar   } }</pre>
--	---	--

**Escuela de Ingeniería**

**Parcial de: Programación I**

**Código de materia:**

**Fecha: Diciembre 2020**

**Grupo: MATUTINO**

**Hoja 2 de 2**

**v11**

**2a) (máximo 5 puntos)** En la clase Socio, implementar un método que dados 2 días y un deporte (1-20), retorne la cantidad de reservas de ese deporte que tiene entre esos 2 días (ambos inclusivos). Se asume desde <= hasta.

**firma: cantidadReservas(desde, hasta, deporte)**

**2b) (máximo 5 puntos)** En la clase Sistema, retornar la lista de socios (ordenada por nombre de socio) con los socios que tienen más de una cantidad dada de reservas (de cualquier fecha y deporte).

**firma: sociosConMásDe(cantidad)**

**2c) (máximo 5 puntos)** Asumir disponible la página de reservas del club. Anotar el código HTML a agregar en dicha página para incluir una section que contenga un formulario con un campo numérico (entre 1 y 40) y un botón con id: "idConsulta". Dentro de la section, pero fuera del formulario habrá una lista ordenada (que se carga en el siguiente punto). Anotar también el CSS a incluir en estilos.css para que esa section tenga color de letra azul.

**2d) (máximo 6 puntos)** Al presionar el botón se validará el formulario y se cargará en la lista ordenada, la lista de socios con más de la cantidad indicada en el campo numérico (asumiendo disponible el método de 2b). De cada socio mostrar el nombre. En el archivo fuentes.js se tiene:

```
window.addEventListener("load", inicio);
let sistema = new Sistema(); // contiene la instancia de Sistema
function inicio(){
    document.getElementById("idConsulta").addEventListener("click", consulta);
}
// parte 2d)
function consulta(){
    // codigo a completar
}
```