

Ejercicios estilo escrito

Tenistas (parcial mat set 2019) - Nivel básico

Realizar un programa en JS que permita obtener estadísticas de un torneo de tenis en el que participaron más de 50 jugadores. Se juega a dos rondas y el puntaje total de cada jugador es la suma de sus dos rondas. De cada participante, se ingresará el nombre, el puntaje que obtuvo en la primera ronda y el puntaje que obtuvo en la segunda. No hay datos incorrectos. El fin de ingreso está dado por el nombre: "FIN" y ambos puntajes en 0. Al final, indicar cuántos jugadores mejoraron su desempeño de la primera a la segunda ronda (o sea, tuvieron mejor puntaje en la segunda ronda), cuál fue el puntaje total máximo del torneo, nombre de un jugador que obtuvo ese puntaje total máximo (si hubiera varios, cualquiera de ellos) y cantidad de jugadores que obtuvieron ese puntaje total máximo.

Dieta (parcial mat set 2019) - Nivel medio

En una cierta dieta, cada día se debe comer solamente alimentos de los indicados, sin importar las cantidades. Cada alimento está representado por una letra mayúscula diferente. Realizar un programa en JS que solicite 3 strings: el primero representa la dieta, el segundo lo ingerido en el almuerzo y el tercero lo ingerido en la cena e indique si ese día se cumplió con la dieta o no. Se asume se ingresarán en mayúsculas.

Ejemplos:

Dieta: "BNACDE"	Dieta: "BTEAG"
Almuerzo: "AEF"	Almuerzo: "GGGGGT"
Cena: "BBCABJ"	Cena: "BA"
Mostrar: No se cumplió	Mostrar: Se cumplió

Repartos (parcial mat set 2019) - Nivel avanzado

Un camión hace repartos. Debe entregar una cierta cantidad de paquetes, pero no puede superar el peso máximo en cada viaje. Deben entregarse en el mismo orden que se reciben.

Ningún paquete supera la carga máxima del camión. Implementar una función en JS que reciba el peso máximo y la cantidad de paquetes como parámetro. Dentro de la función se debe solicitar el ingreso del peso de cada uno de esos paquetes y retornar la cantidad mínima de viajes que debe hacer.

Agregar la invocación por consola a la función para llevar 5 paquetes en un camión de 100 kilos.

```
function viajes(cantidadPaquetes, cargaMaxima)
```

Ejemplos:

Se recibe como cantidad 3 paquetes y la carga máxima 10. Los paquetes que se ingresaron son: 3 2 3, debe retornar 1.

Se recibe 6 paquetes y la carga máxima es 10. Los paquetes ingresados son: 5 4 3 2 1 8, debe retornar 3.

Armónico (parcial especial mayo 2019) – Nivel medio

Un string es armónico si la cantidad de letras “a” en posiciones pares es igual a la cantidad de letras “b” en posiciones impares.

Ej. “AfgBmbasAabxb” no es armónico ya que hay 3 letras A en posiciones pares (posiciones: 0,6,8) y 2 letras “B” en posiciones impares (posiciones: 3,5).

Hacer un programa en JS que solicite un string e indique si es armónico o no.

Club (parcial especial mayo 2019) – Nivel avanzado

En un club se asigna como contraseña a cada socio un string generado de la siguiente forma: comienza con las letras “CA” (contraseña asignada) y luego 18 dígitos, los cuales inician con ceros (eventualmente ninguno) y terminan con el número de socio particular.

Ejemplos:

CA000000000000000012340 es del socio 12340

CA123456789012345678 es del socio 123456789012345678

CA00000000000000000100 es del socio 100

CA12345 es inválida

12345678901234567890 es inválida

Implementar una función que recibe la contraseña y retorna el número de socio particular o el aviso de que no es válida. Ejemplificar la invocación con la contraseña CA12345

```
function proceso(contraseña)
```

Patrón contraseñas (parcial mayo 2019 especial) – Nivel avanzado

Dos contraseñas tienen el mismo patrón cuando la diferencia entre cada uno de sus dígitos posición a posición es la misma. Se asumen del mismo largo.

Ejemplo: contraseñas 2406 y 5739 tienen el mismo patrón:

2	4	0	6
5	7	3	9

Explicación: 2 y 5 difieren en +3, 4 y 7 difieren en +3, 0 y 3 difieren en +3, 6 y 9 difieren en +3
Ejemplo: 75920 y 64831 no tienen el mismo patrón.

Implementar en JS la función verificar que recibe dos contraseñas y retorna el valor booleano correspondiente. Anotar la invocación para probar en la consola con las contraseñas 2402 y 5523.

```
function patron (clave1, clave2)
```

Whatsapp (parcial setp 2019 noct.) – Nivel medio

En un grupo de Whatsapp se agregaron varias personas (representadas por su inicial en mayúscula, sin repetidos) y se dieron de baja otras. Se tiene un string con la secuencia de agregados/eliminados. El signo “+” indica que se agregó la siguiente persona. El signo “-” indica que se bajó la siguiente persona. Ejemplo: +A+C-A+D se agregaron A, C y D y se bajó A, el grupo queda con C y D.

Hacer un programa en JS que lea la secuencia de movimientos e indique los integrantes finales del grupo (en cualquier orden). Inicialmente el grupo está vacío. No hay inconsistencias.

Ejemplo

Se ingresa: +F+H-H+G-F-G

Se muestra: Grupo vacío

Se ingresa: +A+C+N-C+B+C

Se muestra: ACNB (en ese u otro orden)

Lluvia (parcial noct. Mayo 2019) – Nivel básico

Realizar un programa en JS que solicite al usuario 30 datos. Cada dato corresponde a la cantidad de lluvia de un día (se asume se ingresarán en forma ordenada y correlativa del día 1 al día 30). Si un dato de lluvia está fuera del rango 0 a 300, debe solicitarse nuevamente hasta que esté correcto. Luego de ingresar todos los datos, indicar qué día tuvo la máxima cantidad de lluvia (si hubiera varios con el mismo valor, el más cercano al comienzo del mes) y la cantidad de días en que se dio ese máximo.

Máquina (parcial noct. Mayo 2019) – Nivel medio

Una máquina recibe una secuencia de tareas. Realiza dos tipos de trabajo (“A” y “B”). El trabajo “A” se realiza de a uno por vez y lleva siempre una unidad de tiempo. Cada trabajo de tipo “B”, se realiza de a uno por vez y si está inmediatamente precedido por un trabajo “A” lleva 3 unidades de tiempo y sino lleva 2 unidades. Se sabe que la primera tarea siempre es “A”.

Hacer un programa en JS que solicite el string que representa la secuencia (se asume válida) y muestre el tiempo total de trabajo.

Ejemplos

secuencia: "AABABBBAB" muestra: 15

secuencia: ABBB muestra: 8

Bicicletas (parcial noct. Mayo 2019) – Nivel avanzado

En una ciudad, la intendencia desea que todas las bicicletas tengan chapa con 3 dígitos. A efectos de visualizarlas mejor, se trata que ninguna chapa tenga dígitos repetidos, así por ejemplo, la placa 848 no sería válida. Se desea saber la cantidad de chapas que se pueden fabricar entre dos números dados. Implementar en JS la función placas que recibe los dos números y retorna la cantidad.

Ejemplos: Si recibe 105 y 127 retorna 11 (explicación: corresponde a las chapas 105, 106, 107, 108, 109, 120, 123, 124, 125, 126 y 127, en total 11.)

Si recibe 989 y 999 retorna 0

Si recibe 122 y 125, retorna 3

Si recibe 234 y 234, retorna 1

Agregar la invocación a la función para que muestre el resultado para el caso de 105 y 127 utilizando alert.

```
function placas(desde, hasta)
```

Eventos (parcial mat. Mayo 2019) – Nivel básico

Realizar un programa en JS para eventos. Se ingresarán datos de eventos hasta que el usuario indique la palabra "fin". De cada evento se solicita al usuario su tipo ("público" o "privado") y la cantidad de personas que asistieron a ese evento particular. Se asume que todos los datos ingresados son válidos. Al final, indicar cuál tipo de evento tuvo en total más participantes o si ambos tipos tuvieron la misma cantidad total. Si no hubo ningún dato, indicarlo. Además, si hubo datos, informar el promedio de asistencia de personas en los eventos de tipo "privado".

String válido (parcial mat. Mayo 2019) – Nivel medio

Leer un string que representa un párrafo. Indicar si es válido. Se considera válido si tiene más de 20 caracteres y cada vez que hay una coma, el siguiente carácter es un espacio.

Ejemplos:

Se ingresa: "Este es el texto. Para revisar, mañana sin falta." Es válido

Se ingresa: "Esta frase aunque larga, no es ok, opss" No es válida

Se ingresa: "Frase larga y también incorrecta," No es válida