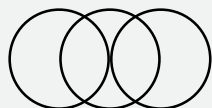
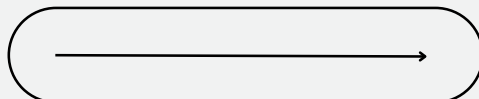


# TEAM



# TRAINING BOOK

8

## \* CALCULADORA .TXT

8

\* Reto #8

\* Dificultad: MEDIA

\* Calcula su resultado e imprímelo.

\* Enunciado: Crear el fichero "Challenge21.txt" e incluyelo en el proyecto, por ejemplo:

5

+

2

-

1

\*

8

-

15

+

4

/

2

\* Calcula su resultado e imprímelo.

\* - El .txt se corresponde con las entradas de una calculadora.

- \* - Cada línea tendrá un número o una operación representada por un símbolo (alternando ambos).
- \* - Soporta números enteros y decimales.
- \* - Soporta las operaciones suma "+", resta "-", multiplicación "\*" y división "/".
- \* - El resultado se muestra al finalizar la lectura de la última línea (si el .txt es correcto).
- \* - Si el formato del .txt no es correcto, se indicará que no se han podido resolver las operaciones.

9

## \* BATALLA POKÉMON

9

- \* Reto #9
- \* Dificultad: MEDIA
- \*
- \* Enunciado: Crea un programa que calcule el daño de un ataque durante una batalla Pokémon.
- \* - La fórmula será la siguiente:  $\text{daño} = 50 * (\text{ataque} / \text{defensa}) * \text{efectividad}$
- \* - Efectividad: x2 (súper efectivo), x1 (neutral), x0.5 (no es muy efectivo)
- \* - Sólo hay 4 tipos de Pokémon: Agua, Fuego, Planta y Eléctrico (buscar su efectividad)
- \* - El programa recibe los siguientes parámetros:
- \* - Tipo del Pokémon atacante.
- \* - Tipo del Pokémon defensor.
- \* - Ataque: Entre 1 y 100.
- \* - Defensa: Entre 1 y 100.

10

## \* CICLO SEXAGENARIO CHINO

10

- \* Reto #10
- \* Dificultad: MEDIA

\*

\* Enunciado: Crea un función, que dado un año, indique el elemento y animal correspondiente en el ciclo sexagenario del zodiaco chino.

\* - Información:

<https://www.travelchinaguide.com/intro/astrology/60year-cycle.htm>

\* - El ciclo sexagenario se corresponde con la combinación de los elementos madera,

\* fuego, tierra, metal, agua y los animales rata, buey, tigre, conejo, dragón, serpiente,

\* caballo, oveja, mono, gallo, perro, cerdo (en este orden).

\* - Cada elemento se repite dos años seguidos.

\* - El último ciclo sexagenario comenzó en 1984 (Madera Rata).

## **11 \* ITERATION MASTER**

**11**

\* Reto #11

\* Dificultad: FÁCIL

\*

\* Enunciado: Quiero contar del 1 al 100 de uno en uno (imprimiendo cada uno). ¿De cuántas maneras eres capaz de hacerlo? Crea el código para cada una de ellas.

## **12 \* TRUCO O TRATO**

**12**

\* Reto #12

\* Dificultad: FÁCIL

\*

\* Enunciado: Este es un reto especial de Halloween.

\* Deberemos crear un programa al que le indiquemos si queremos realizar "Truco o Trato" y

- \* un listado (array) de personas con las siguientes propiedades:
- \* - Nombre de la niña o niño
- \* - Edad
- \* - Altura en centímetros
- \*
- \* Si las personas han pedido truco, el programa retornará sustos (aleatorios)
- \* siguiendo estos criterios:
- \* - Un susto por cada 2 letras del nombre por persona
- \* - Dos sustos por cada edad que sea un número par
- \* - Tres sustos por cada 100 cm de altura entre todas las personas
- \* - Sustos: 🎃 👻 💀 🕷️ 🕸️ 🦇
- \*
- \* Si las personas han pedido trato, el programa retornará dulces (aleatorios)
- \* siguiendo estos criterios:
- \* - Un dulce por cada letra de nombre
- \* - Un dulce por cada 3 años cumplidos hasta un máximo de 10 años por persona
- \* - Dos dulces por cada 50 cm de altura hasta un máximo de 150 cm por persona

<b>13</b>	<ul style="list-style-type: none"> <li>* <b>¿DÓNDE ESTÁ EL ROBOT?</b></li> <li>* Reto #13</li> <li>* Dificultad: MEDIA</li> <li>*</li> <li>* Enunciado: Calcula dónde estará un robot (sus coordenadas finales) que se encuentra en una cuadrícula representada por los ejes "x" e "y".</li> <li>* - El robot comienza en la coordenada (0, 0).</li> </ul>	<b>13</b>
-----------	--	-----------

- \* - Para indicarle que se mueva, le enviamos un array formado por enteros (positivos o negativos)
- \* que indican la secuencia de pasos a dar.
- \* - Por ejemplo: [10, 5, -2] indica que primero se mueve 10 pasos, se detiene, luego 5, se detiene,
- \* y finalmente 2. El resultado en este caso sería (x: -5, y: 12)
- \* - Si el número de pasos es negativo, se desplazaría en sentido contrario al que está mirando.
- \* - Los primeros pasos los hace en el eje "y". Interpretamos que está mirando hacia la parte
- \* positiva del eje "y".
- \* - El robot tiene un fallo en su programación: cada vez que finaliza una secuencia de pasos gira
- \* 90 grados en el sentido contrario a las agujas del reloj.

14

#### \* MÁQUINA EXPENDEDORA

14

- \* Reto #14
- \* Dificultad: MEDIA
- \*
- \* Enunciado: Simula el funcionamiento de una máquina expendedora creando una operación
- \* que reciba dinero (array de monedas) y un número que indique la selección del producto.
- \* - El programa retornará el nombre del producto y un array con el dinero de vuelta (con el menor
- \* número de monedas).
- \* - Si el dinero es insuficiente o el número de producto no existe, deberá indicarse con un mensaje
- \* y retornar todas las monedas.
- \* - Si no hay dinero de vuelta, el array se retornará vacío.

\* - Para que resulte más simple, trabajaremos en céntimos con monedas de 5, 10, 50, 100 y 200.

\* - Debemos controlar que las monedas enviadas estén

---

All rights reserved. No part of this script may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the copyright owner.