



Ejercicio 1:

Crea un *array* con 50 números enteros aleatorios. A partir del mismo, crea una cadena de respuesta que cuando encuentre que alguno de ellos es un número primo lo incluya. El resultado ha de ser una cadena similar a: *"Los números primos aleatorios de esta lista son x, x, x, x, x"*.

Ejercicio 2:

Realiza una función que analice un texto y devuelva el número de palabras del mismo, su longitud completa, y qué longitud tiene cada palabra que aparezca en dicho texto en un listado.

Ejercicio 3:

Crea una fábrica de chocolate con clases. La clase principal hará chocolatinas de *n* onzas por tableta (*n* es aleatorio). Las tabletas pueden ser de chocolate negro, con leche o blanco. En cada tableta además podría venir un billete dorado para visitar la fábrica, pero en una probabilidad muy baja.

Ejercicio 4:

Vamos a crear un colegio donde crearemos la estructura de datos necesarios usando *structs* en vez de clases.

- *Struct* para los **Profesores**, con nombre y edad.
- *Struct* para las **Asignaturas**, con nombre de asignatura y cursos donde se imparte.
- *Struct* para los **Alumnos**, con nombre, edad y curso.
- *Struct* para los **Cursos** con el tutor (profesor) y los alumnos.
- *Struct* del **Colegio** con los distintos cursos del mismo.

Queremos poder extraer listados y/o conteo de las asignaturas que imparte cada profesor, cuántos alumnos tiene cada uno y cuántos alumnos distintos hay en el colegio que al menos estén en una asignatura.

Para probar que funciona crea una serie de datos de prueba.

Ejercicio 5:

Vamos a crear la estructura de un videojuego.

- **Personaje** tiene un valor de vida y un nombre. Es capaz de morir cuando se queda sin vida y de saludar diciendo su nombre.
- **Héroe** hereda de **Personaje**. Tiene valor de fuerza de ataque y es capaz de atacar a un **Enemigo** que recibe como parámetro, reduciendo su vida en un ataque igual a un número entre 0 y la mitad de su fuerza.
- **Mago** hereda de **Héroe**. Tiene un valor de magia que suma la mitad de su valor al daño ocasionado en un ataque y reduce en 1/4 el daño infligido cuando es atacado por un **Enemigo**.
- **Guerrera** hereda de **Héroe**. Tiene una espada que en su inicialización tendrá un valor de ataque que será un número aleatorio de 0 a la mitad

de su fuerza. Dicha espada inflige un daño extra a los enemigos igual al total de dicha fuerza.

- **Enemigo** hereda de **Personaje**. Tiene valor de fuerza de ataque y puede tener tres tipos de armas, la cual se elegirá en su inicialización. Puede atacar a un héroe, recibido como parámetro. El arma **Hacha** hará un daño de 10 sobre el daño del ataque, el arma **Sable** lo hará de 5 y el arma **Cadena** lo hará de 2. También puede ser que el **Enemigo** se cree sin arma y entonces no aumente su fuerza en el ataque.

Crear todos los inicializadores, métodos y propiedades necesarios, así como la capacidad de atacar y morir de cada personaje susceptible de ello, además del resto de comportamientos indicados. Usa enumeraciones donde sea más práctico para representar datos.

Ejercicio 6:

Enviada a una función un *array* de números enteros, devuelve un diccionario en cuya clave esté cada número único en dicho array y en el valor las veces que aparece consecutivamente dentro del *array* enviado.

Ejercicio 7:

Enviada una frase a una función, descomponga esta en sus distintas palabras. Una vez hecho, devuelva un diccionario con las palabras como clave y en el valor, el número de veces que aparece cada palabra en la frase.