Vectores y matrices
Relación de ejercicios

1. Dado un vector de números reales:

    a. Escriba un método `max` que nos devuelva el máximo de los valores incluidos en el vector.

    b. Escriba un método `min` que nos devuelva el mínimo de los valores incluidos en el vector.

    c. Escriba un método `media` que nos devuelva la media de los valores incluidos en el vector.

    d. Escriba un método `varianza` que nos devuelva la varianza de los valores incluidos en el vector.

    e. Escriba un método `mediana` que nos devuelva la mediana de los valores incluidos en el vector.

    f. Escriba un método `moda` que nos devuelva la moda de los valores incluidos en el vector

    g. Escriba un método `percentil(n)` que nos delvuelva el valor correspondiente al percentil n en el conjunto de valores del vector.

2. Implemente una clase en Java, llamada `Serie`, que encapsule un vector de números reales e incluya métodos (no estáticos) que nos permitan calcular todos los valores mencionados en el ejercicio anterior a partir de los datos encapsulados por un objeto de tipo `Serie`.

3. Dado un vector de números reales, escriba un método que nos devuelva el máximo y el mínimo de los valores incluidos en el vector.

4. Dado un vector, implemente un método que inserte un elemento en una posición dada del vector.

    NOTA: Insertar un elemento en el vector desplaza una posición hacia la derecha a los elementos del vector que han de quedar detrás del elemento insertado. Además, la inserción ocasiona la “desaparición” del último elemento del vector.

5. Implemente un método llamado secuencia que realice la búsqueda de la secuencia en orden creciente más larga dentro de un vector de enteros. El método ha de devolver tanto la posición de la primera componente de la secuencia como el tamaño de la misma.

6. Una cadena de ADN se representa como una secuencia circular de bases (adenina, timina, citosina y guanina) que es única para cada ser vivo, por ejemplo:

A	T	G
T		C
A	T	G

   Dicha cadena se puede representar como un vector de caracteres recorriéndola en sentido horario desde la parte superior izquierda:

A	T	G	C	G	T	A	T

   Se pide diseñar una clase que represente una secuencia de ADN e incluya un método booleano que nos devuelva `true` si dos cadenas de ADN coinciden.

   MUY IMPORTANTE: La secuencia de ADN es cíclica, por lo que puede comenzar en cualquier posición. Por ejemplo, las dos secuencias siguientes coinciden:

A	T	G	C	G	T	A	T

A	T	A	T	G	C	G	T

7. Dado un vector de números reales, escriba un método que ordene los elementos del vector **de mayor a menor**.

8. Dado un vector de números reales, escriba un método que ordene los elementos del vector de tal forma que los números pares aparezcan antes que los números impares. Además, los números pares deberán estar ordenados de forma ascendente, mientras que los números impares deberán estar ordenados de forma descendente. Esto es, el vector {1,2,3,4,5,6} quedará como {2,4,6,5,3,1}.

9. Crear una clase `Matriz` para manipular matrices que encapsule un array bidimensional de números reales.

   a. Incluya en la clase métodos que nos permitan acceder y modificar de forma segura los elementos de la matriz (esto es, las variables de instancia deben ser privadas y los métodos han de comprobar la validez de sus parámetros).

   b. Escriba un método que nos permita sumar matrices.

   c. Implemente un método que nos permita multiplicar matrices.

   d. Cree un método con el que se obtenga la traspuesta de una matriz.

10. En Java, para generar números pseudoaleatorios, se puede utilizar la función `Math.random()` definida en la clase `java.lang.Math`. Dicha función genera una secuencia de números pseudoaleatorios que se supone sigue una distribución uniforme (esto es, todos los valores aparecerán con la misma probabilidad). Escriba un programa que compruebe si el generador de números pseudoaleatorios de Java genera realmente números aleatorios con una distribución uniforme.

    Sugerencia: Genere un gran número de números aleatorios (entre 0 y 100, por ejemplo) y compruebe que la distribución resultante del número de veces que aparece cada número es (más o menos) uniforme. Por ejemplo, mida la dispersión de la distribución resultante utilizando una medida como la varianza (y reutilice la clase `Serie` del ejercicio 2).

11. Realizar una simulación de Monte Carlo para aproximar el valor del área bajo una curva f(x), en $x \in [a, b]$.

    Algoritmo: Generar puntos aleatorios en el rectángulo de extremos (a, 0) y (b, m), con $m = \max_{a \le x \le b} f(x)$, y contar el número de puntos que caen por debajo de la curva.

    NOTA: Este método permite generar números pseudoaleatorios que sigan cualquier distribución que nosotros deseemos. Por ejemplo, probar con una distribución normal.

12. Crear un programa modular para jugar a las **7 y media**. Se trata de un juego de cartas (con baraja española) en el que el objetivo es alcanzar una puntuación de 7.5. Cada carta del 1 al 7 tiene su valor nominal y cada figura (sota, caballo y rey) vale 0.5 puntos.

    NOTA: Para barajar, mezcle los elementos de un vector de cartas intercambiando en repetidas ocasiones cartas elegidas al azar con la ayuda de la función `Math.random()`