

Instrucciones:

- La fecha de entrega es el día: Jueves 2 de septiembre a las 8am.
- Cada día de atraso implicará una pérdida de 20 puntos.
- Las tareas son estrictamente de carácter individual, tareas iguales se les asignará cero puntos.
- Deben incluir 3 casos de prueba por cada ejercicio.

## TAREA NÚMERO 2

1. [5 puntos] Desarrolle una función recursiva que reciba un vector de números en una lista y un número  $x$ , tal que retorne el porcentaje de elementos menores o iguales a  $x$ .
2. [15 puntos] Escriba funciones recursivas para las siguientes operaciones con listas:
  - (Aplica  $f$   $L$ ): Aplica una función  $f$  a toda la lista  $L$ .
  - (Ordena  $L$ ): ordena la lista dada por parámetro.
  - (Suma3  $L1$   $L2$   $L3$ ): suma tres listas y el resultado lo deja en una cuarta lista.
  - (Subconjunto?  $L1$   $L2$ ): retorna TRUE si la lista  $L1$  está contenida en la lista  $L2$ .
  - (eliminar  $x$   $L$ ): elimina la primera aparición de  $x$  en la lista  $L$ , el resultado lo deja en otra lista.
3. [15 puntos] Defina un sistema para la aritmética de los números complejos de la forma  $a + bi$ , para esto las operaciones algebraicas se definen como sigue:
  - **Suma:**  $(a + bi) + (c + di) = (a + c) + (b + d)i$
  - **Resta:**  $(a + bi) - (c + di) = (a - c) + (b - d)i$
  - **Producto:**  $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$
  - **Conjugado:**  $\overline{(a + bi)} = a + (-b)i$
  - Se debe definir además una función que despliegue en pantalla un número complejo de la forma  $(a + bi)$ .
4. [10 puntos] Construya una función recursiva (**divisibles**  $L$   $x$ ) que retorne una lista compuesta por dos sublistas que contienen los elementos de la lista original, en la primera de las listas están los elementos que son divisibles por  $x$  y en la segunda los elementos que no.
5. [10 puntos] Construya una función (**pertenece?**  $x$   $L$ ) que retorne verdadero o falso dependiendo de si el elemento  $x$  pertenece a la lista  $L$  o no. Utilizando la función **pertenece?** construya una función (**resta**  $L1$   $L2$ ) que le resta a la lista  $L1$  la lista  $L2$ , retornando una lista con aquellos elementos que están en  $L1$  pero no en  $L2$ . Por ejemplo:  
(resta '(7 1 4 3 2 4 2 4 6) '(5 1 4)) =>(7 3 2 2 6)

6. [10 puntos] Construya una función recursiva que recibe 2 listas de tamaño  $n$  y retorna una lista con  $n$  pares ordenados (listas) como se ilustra en el siguiente ejemplo:

(Pares '(7 1 4 3) '(5 1 4 -1)) =>((7 5) (1 1) (4 4) (3 -1))

7. [15 puntos] Desarrolle una función recursiva que reciba un valor y retorne una lista con los factores primos de dicho valor. Los factores primos de un número entero son los números primos divisores exactos de ese número entero. El proceso de búsqueda de esos divisores se denomina factorización de entera, o factorización en números primos. Por ejemplo, para 60 el proceso es el siguiente:

- $60/2 = 30$ .
- $30/2 = 15$ .
- $15/3 = 5$ .
- $5 = 5 = 1$ .
- Por tanto, el resultado es: (2 2 3 5).

8. [20 puntos] Suponga que una terminal de buses quiere desarrollar un sistema para devolver la cantidad mínima de monedas para dar vuelto a un pasajero. Es decir, quiere que el vuelto dado a un pasajero sea con la mínima cantidad de monedas. Por tanto, desarrolle un sistema que reciba el vuelto, por teclado, que se le debe dar al pasajero y retorne una lista con las monedas que se le deben dar a dicho pasajero. Las monedas a utilizar son: C1, C5, C10, C25, C50, C100 y C500. El sistema debe incluir lo siguiente:

- a) **es-menor?**: determina si un valor es menor a otro. Esto nos sirve para indicar si una moneda se puede utilizar o no dependiendo del vuelto a dar.
- b) **vuelto**: obtiene la lista de monedas a dar dado el valor del vuelto. Debe programarla con recursión o recursión lineal.
- c) **resultado**: debe de leer de teclado el valor del vuelto y despliega la lista con las monedas a dar.



**oldemar** **rodríguez**  
CONSULTOR en MINERÍA DE DATOS