

- Problema 1)** a) Marque y explique los errores en el siguiente fragmento de código. (+1.5)  
b) Escriba el código con los errores corregidos, conservando la lógica del problema. (+1.0)

```

struct Point {
    int x, int y;
};
void pointxy_difference(struct p1, p2){
    p2.x -= p1.x;
    p2.y -= p1.y;
    return p2;
}

int main( ){
    struct Point p1 = { 0, 0 };
    struct Point p2;
    if (p1 == p2)
        printf("Los puntos son iguales\n");
    p1 = pointxy_difference( p1, p2);
    printf("diferencia x=%d, y=%d\n", p1 );
    return 0;
}

```

- Problema 2)** Suponga una estructura para modelar información de los autos que circulan en alguna ciudad, cuya definición es:

```

struct automovil {
    char placas[10];
    char marca[20];
    char modelo[20];
    int serie;
};

```

- a) **Escriba una función** que reciba un arreglo de una dimensión con información de automóviles usando la estructura previamente definida y la cantidad de elementos en el arreglo. Ordene de manera ascendente los automóviles usando el **algoritmo bubble sort (+2.0)**, considere como criterio de ordenamiento el campo que contiene el número de serie (**serie**).

- b) **Escriba una función** que reciba un arreglo de una dimensión con información de varios automóviles (suponga que dicho arreglo está ordenado en base al número de serie), un número de serie a buscar y la cantidad de elementos que existen en el arreglo. Escriba una función que **use búsqueda binaria (+2.0)** o **búsqueda secuencial (+1.0)** para indicar el índice en el arreglo en donde está localizado el automóvil con el número de serie consultado. Regrese como resultado **-1** en caso de que el número de serie no esté en el arreglo.

- Problema 3)** Considere la siguiente tabla de propiedades de la leche almacenada en un archivo binario:

Animal	Proteínas %	Grasas %	Lactosa %
Toro	3.42	4.09	4.82
Bufalo	4.38	7.73	4.79
Oveja	5.73	6.99	4.75
Chivo	3.26	4.07	4.51
Camello	3.26	3.80	4.30
Alpaca	3.90	3.20	5.60

- a) Escriba una estructura para modelar las propiedades de la leche de cada animal, llámela **L**. (+0.5).  
b) Escriba una función que reciba un vector de datos de tipo **L**, el nombre del archivo y otros parámetros que considere. La función debe guardar en el vector todos los elementos de tipo **L**. Suponga que no sabe cuántos registros hay en el archivo\*. (+1.0).  
c) Escriba una función que calcule la distancia euclidiana entre dos elementos de tipo **L**. (+1.0).

$$\text{Dist} = \text{raíz} \left( (x.p_1 - y.p_1)^2 + (x.p_2 - y.p_2)^2 + (x.p_3 - y.p_3)^2 \right)$$

- d) Escriba una función que reciba un vector de elementos de tipo **L**, un elemento **x** de tipo **L**, y regrese el índice del animal más cercano a **x**. (+1.0).  
e) Escriba un programa que cargue los datos de la leche de los animales desde un archivo recibido como parámetro en la función *main*, y los guarde en un vector de tipo **L**. Tome los siguientes 3 parámetros recibidos por la función *main* (para las tres propiedades de la leche), mismas que guardará en una variable **w** (estructura de tipo **L**). Busque el índice del animal cuya distancia es menor a **w**, e indique a qué animal es más parecida la muestra ingresada (+2.0).

**./programa datosLeche.bin 4.379 7.7 4.79**

\*El archivo contiene únicamente los datos, las cabeceras de la tabla se ponen para aclarar el problema y corresponden a las propiedades de la leche. Como máximo habrán datos de 20 animales.