

1. ** Implementar la función `int busqueda_binaria(int x, int numeros[], int n)` que, suponiendo que el arreglo de números está ordenado, devuelve la posición en la que se encuentra `x`, o -1 si no se encuentra.
2. *** Un archivo binario contiene un listado de ciudades, cada una con su ubicación (latitud y longitud). En el archivo, cada ciudad tiene el siguiente formato:

```
++-----+-----+
|N|nombre    |lat |long|
++-----+-----+
```

- `N` es un entero sin signo de 8 bits que representa la cantidad de caracteres de `nombre` (sin incluir el caracter nulo).
- `nombre` es el nombre de la ciudad (sin incluir el caracter nulo).
- `lat` y `long` son dos `float`.

Se pide:

- a. Declarar una estructura para guardar la información de una ciudad.
 - b. Escribir la función `leer_ciudad` que reciba un archivo abierto (tipo `FILE *`), lea del archivo la información de una ciudad y devuelva una instancia (reservando memoria dinámica) de la estructura declarada previamente. Todos los recursos (archivos y memoria dinámica) deben haber sido liberados al finalizar la ejecución, haya ocurrido un error o no.
3. ** Dada la función escrita en el ejercicio 2, escribir un programa que reciba como parámetro de línea de comandos el nombre del archivo, lea del mismo la información de todas las ciudades que contiene y las imprima en la salida estándar.
 4. *** En una lista doblemente enlazada, cada nodo contiene una referencia a los nodos anterior y próximo.

Dado el TDA lista enlazada implementado como:

```
typedef struct nodo {
    int dato;
    struct nodo *prox;
    struct nodo *ant;
} nodo_t;
```

```
typedef struct {
    nodo_t *prim;
} lista_t;
```

Implementar la función `void eliminar(lista_t *lista, int posicion)`.

1. ** Implementar la función `int busqueda_binaria(int x, int numeros[], int n)` que, suponiendo que el arreglo de números está ordenado, devuelve la posición en la que se encuentra `x`, o -1 si no se encuentra.
2. *** Un archivo binario contiene un listado de ciudades, cada una con su ubicación (latitud y longitud). En el archivo, cada ciudad tiene el siguiente formato:

```
++-----+-----+
|N|nombre    |lat |long|
++-----+-----+
```

- `N` es un entero sin signo de 8 bits que representa la cantidad de caracteres de `nombre` (sin incluir el caracter nulo).
- `nombre` es el nombre de la ciudad (sin incluir el caracter nulo).
- `lat` y `long` son dos `float`.

Se pide:

- a. Declarar una estructura para guardar la información de una ciudad.
 - b. Escribir la función `leer_ciudad` que reciba un archivo abierto (tipo `FILE *`), lea del archivo la información de una ciudad y devuelva una instancia (reservando memoria dinámica) de la estructura declarada previamente. Todos los recursos (archivos y memoria dinámica) deben haber sido liberados al finalizar la ejecución, haya ocurrido un error o no.
3. ** Dada la función escrita en el ejercicio 2, escribir un programa que reciba como parámetro de línea de comandos el nombre del archivo, lea del mismo la información de todas las ciudades que contiene y las imprima en la salida estándar.
 4. *** En una lista doblemente enlazada, cada nodo contiene una referencia a los nodos anterior y próximo.

Dado el TDA lista enlazada implementado como:

```
typedef struct nodo {
    int dato;
    struct nodo *prox;
    struct nodo *ant;
} nodo_t;
```

```
typedef struct {
    nodo_t *prim;
} lista_t;
```

Implementar la función `void eliminar(lista_t *lista, int posicion)`.