

Strings y Búsqueda Binaria

—

Ejercicio 1

(cualquier parecido con la realidad es pura coincidencia)

Lisa tiene clase de computación hoy y para eso, van a los laboratorios.

Como las compus son un poco viejas, Lisa quiere buscar la que esté en mejor estado para poder hacer su tarea de la mejor forma.

Dado el siguiente struct:

```
typedef struct tecla {
    char descripción[MAX_NOMBRE]; // "A", "Ctrl", "Insert", etc
    bool esta_rota;
} tecla_t;

typedef struct compu {
    tecla_t teclado[MAX_TECLAS];
    int cantidad_teclas;
    int id;
} compu_t;

typedef struct laboratorio {
    char nombre[MAX_NOMBRE];
    compu_t computadoras[MAX_COMPUS];
    int cantidad_compus;
} laboratorio_t;
```

- a. Crear una función que encuentre la computadora a la que le ande la mayor cantidad de teclas y que además le ande "shift" y "ctrl".
- b. Bart quiere hacer una broma en la escuela cambiando el laboratorio llamado "Skinner" a "Baboso". Sabiendo que los laboratorios están ordenados por orden alfabético, armar un procedimiento que realice este cambio.

Ejercicio 2 (de parcial)

¡El Profesor Frink ha hecho un descubrimiento increíble, encontró un posible planeta B, el cual nombró Frinkon-7!

Para confirmar su hallazgo ha realizado mediciones de oxígeno en la atmósfera para poder determinar si el planeta es habitable.

Para que el planeta sea habitable, el nivel de oxígeno de todas sus mediciones **estables** debe ser mayor a 100.

Las mediciones espaciales están guardadas en un vector de tipo **medicion_t**:

```
typedef struct medicion {  
    char planeta[MAX_NOMBRE];  
    bool es_estable;  
    int nivel_oxigeno;  
} medicion_t;
```



Se pide implementar una función **recursiva** que reciba un vector con todas las mediciones de planetas que ha hecho el profesor, su tope y devuelva **true** si el planeta Frinkon-7 es habitable y **false** en caso contrario.