

# Recursividad, Unión, intersección y diferencia entre vectores

## Ejercicio 1

Cuando Lilo no está enseñándole a Stitch, juegan bastante seguido a las escondidas. El problema es que Stitch se esconde muy bien, tanto, que Lilo a veces se cansa de buscarlo y pide ayuda, y ahí es cuando aparecemos nosotros, para darle una mano.

Se puede pensar el campo de juego como una matriz de caracteres, en donde los espacios vacíos están representados con ` ` (espacio) y Stitch está representado con `S`.

1. Implementar una función recursiva que devuelva `true` si se encuentra a Stitch o `false` en caso contrario.

## Ejercicio 2

El sueño de Pinocho era convertirse en un niño de verdad, pero no solo eso, sino también crecer y ver a Messi traer la copa.

Se avecina un mundial más y como cada 4 años, re manijas *Pinocho* y *Pepe Grillo* compiten para ver quien logra llenar antes el álbum de figuritas.

Cada figurita guarda información de un jugador:

```
typedef struct jugador{
    int id;
    char nombre[MAX_NOMBRE];
    int dorsal;
    char pais[MAX_PAIS];
} jugador_t;
```

Y las figuritas de ambos están en un vector con su tope para cada uno:

```
jugador_t pilon_pinocho[MAX_PILON];
int tope_pinocho;
jugador_t pilon_pp_grillo[MAX_PILON];
int tope_pp_grillo;
```

Los vectores están ordenados por el **id** ascendentemente.

En esta oportunidad queremos desarrollar una biblioteca que sirva para realizar acciones entre los pilones de figuritas de *Pinocho* y *Pepe Grillo*.

Con la biblioteca se debe poder:

- Obtener las figuritas que tiene uno y el otro no.
- Obtener las figuritas que tienen ambos en común.
- Obtener las figuritas, sin repetición que juntan entre los dos.
- Obtener las figuritas de los dos en total.

*Aclaración: Siempre se tomará como precondition que los pilones están ordenados y como postcondición se tendrá que el pilón de salida, estará también ordenado.*

## Tarea

Se cuenta con un vector que contiene las letras de una contraseña. La particularidad de este vector es que cada elemento de él contiene el valor de la letra y la posición de la letra que debe ser mostrada.

```
typedef struct celda {
    char letra;
    int posicion;
} celda_t;
```

Si la cantidad está en -1, en esa posición debe mostrarse un espacio.

Por ejemplo, si se tiene el vector:

U 4	Y 2	L 8	N -1	O 6	T 9	K -1	S 8	A 7	E 9
--------	--------	--------	---------	--------	--------	---------	--------	--------	--------

La contraseña será:

**"OLA KE ASE"**

- Crear una rutina recursiva que muestre la contraseña.