

1.- Dado un Array a como el siguiente, únalo al Array b formando el Array c. Determine la cantidad de elementos del array resultante.

```
a = ["Lunes", "Martes", "Miércoles"]
b = ["Jueves", "Viernes", "Sábado", "Domingo"]
```

Considere 2 alternativas diferentes para realizar este ejercicio.

2.- Teniendo en cuenta los siguientes vectores, construya los métodos necesarios para obtener la diferencia entre el primero y el segundo vector, la intersección entre ambos y la unión entre ambos, usando la codificación estándar por usted conocida para arrays. Para cada salida lograda (array resultante), compárelo con la *ruby-way* mostrada en la tabla de más abajo.

```
op_sys = ["Fedora", "SuSE", "Tuquito", "Windows", "MacOS"]
linux_sys = ["Tuquito", "SuSE", "PCLinuxOS", "Ubuntu", "Fedora"]
```

Operador (Ruby-way)	Descripción
-	Diferencia - Retorna un Nuevo array que contiene elementos que están en el primer vector pero no en el segundo.
&	Intersección – Crea un Nuevo array desde 2 existentes arrays conteniendo solo los elementos comunes a ambos arrays. Los duplicados son removidos.
	Unión – Concatena 2 arrays. Duplicados son removidos.

3.- Remueva los elementos duplicados en un array de 100 elementos enteros generados al azar con valores (-999..-1), usando el método tradicional y compare el resultado con el método `unique`.

4.- Invierta los elementos del siguiente array y compárelo con algún método para inversión de arrays que encuentre en Ruby.

```
colores = ["rojo", "verde", "azul", "negro", "celeste", "blanco"]
```

5.- Genere un array con 500 números enteros al azar mayores que 0 y menores que 100. Ordene el vector por un método conocido. Luego realice el ordenamiento por el método `sort` de Ruby. Determine de alguna manera cuál es el método que tarda menos tiempo en realizar el ordenamiento.

6.- Genere un array A al azar con N elementos al azar. (N es también un nro. Al azar), haga lo mismo con un array B (M al azar). Ordene ambos vectores en forma independiente pero con un único `def`. Pase al array C todos los elementos de los vectores sin repetir y en correcto orden ascendente.

7.- Genere un array con 50 números enteros al azar mayores que 0 y menores que 500. Luego realice el ordenamiento por el método `sort` de Ruby. Solicite un dato ingresado por teclado. Escriba una `def` que busque ese dato en el array utilizando el método de *búsqueda binaria* (investigue sobre este tema).

8.- Se tiran 500 veces un dado. Consignar los resultados en un array. Luego generar otro array con los valores que más salieron.

9.- Cargue un array con 10 nombres de personas. Genere un segundo vector con los nombres en mayúsculas y un tercer vector con los nombres al revés.

10.- Cargue un array con 10 notas. Genere el array resultado con el promedio entre la `nota[0]` y la `nota[9]`, `nota[1]` con `nota[8]` y así sucesivamente. Determine el promedio del vector resultado.