Programación II
Práctica 0: Acumuladores booleanos

Como vimos en clase un **acumulador** booleano toma sólo dos valores, true o false. Se denomina acumulador, porque suma un resultado parcial al resultado final de la función. La variable "ret" es el nombre abreviado de "valor de retorno".

Cuantificadores

Cuando queremos probar una propiedad P para todo un conjunto de datos:

$$\{\forall\ x\ \in lista\ /\ P(x)\} \equiv true$$

Utilizaremos la hipótesis de ret = ***true*** y la acumulación será de la forma:

```
ret = ret AND P(x)
```

Cuando queremos probar una propiedad P para un solo elemento:

$$\{\exists\ x\ \in lista\ /\ P(x)\} \equiv true$$

Utilizaremos la hipótesis de ret = ***false*** y la acumulación será de la forma:

```
ret = ret OR P(x)
```

Ejercicios obligatorios:

1) Implementar con acumuladores booleanos una función booleana "mayor10" que recibe una lista de números enteros, y devuelve True si todos los elementos son mayores a 10.

```
boolean mayor10(int[] lista){ …  }
```

2) Implementar con acumuladores booleanos una función booleana "multiplo5AlgunoMayor100" que recibe una lista de números enteros, y devuelve True si todos los elementos son múltiplo de 5 y alguno de ellos es mayor a 100.

```
boolean multiplo5AlgunoMayor10(int[] lista){ …  }
```

3) Implementar una función que determine si un arreglo es subconjunto de otro:

```
public static boolean pertenecenTodos(int[] elems,
                                       int[] arreglo) ...
```

Casos borde a tener en cuenta:

- *elems* está vacío (y la función devuelve verdadero)
- *arreglo* está vacío (y la función devuelve falso)

- alguno de los arreglos contiene duplicados (no influye, es suficiente con que estén una vez)

Algunos ejemplos:

```
[1, 2] ⊆ [3, 2, 1]
[4, 1] ⊈ [1, 2, 3]
[2, 2] ⊆ [1, 2, 3]
```

4) Implementar una clase en Java que tenga métodos estáticos que reciban una matriz por parámetro y muestre los elementos de la matriz por filas y otro que los muestre por columnas, y un tercer método que sume todos sus elementos.
5) Agregar a la clase anterior métodos estáticos que reciban por parámetro una matriz y devuelvan un arreglo con la suma de cada fila de la matriz (cuya dimensión sea la cantidad de filas de la matriz). Crear otro que haga lo mismo, pero con las columnas.
6) Implementar, utilizando acumuladores booleanos, una función que reciba una matriz de enteros, y devuelva verdadero ⇔ en cada una de las filas, existe al menos un número negativo.

```
public static boolean tieneNegativos(int[][] mat){ ...}
```

7) Implementar una función que, dada una matriz de enteros, verifique ambas condiciones:

   a) todas las filas están en orden estrictamente ascendente.
   b) todas las columnas tienen al menos un elemento impar, y otro par.

```
public static Boolean filasCrecientesParImpar(int[][] mat) ...
```

Algunos ejemplos:

```
[[1, 2, 3], [4, 5, 6]] → Verdadero
[[1, 2, 3], [4, 5, 5]] → Falso
[[1, 2, 3], [2, 4, 6]] → Falso
```

8) Implementar funciones con potencias de 2 y logaritmos en base 2 – ES MUY IMPORTANTE QUE PUEDAN HACER ESTE EJERCICIO Y ENTIENDAN LA RELACION ENTRE AMBAS FUNCIONES.

a) Implementar una función que dado un vector de enteros devuelva verdadero ⇔ todos sus elementos son potencia de 2.

Algunos ejemplos:

```
[8, 2, 32] → Verdadero [ 2^3 ,  2^1  ,  2^5 ]
[15, 2, 8] → Falso.   No son todos potencias de 2
```

b) Implementar una función que dada una matriz de enteros devuelva verdadero ⇔ en alguna fila todos sus elementos son potencia de 2. (Usar el punto a)

Algunos ejemplos:

```
[[1, 2, 3], [8, 2, 1]] → Verdadero    fila 1 = [ 2³, 2¹, 2⁰ ]
[[1, 2, 3], [11, 2, 8], [[4, 5, 6]] → Falso ninguna lo cumple
```

c) Implementar una función que dada una matriz de enteros devuelva verdadero ⇔ en alguna fila algún elemento es **Parte Entera($\log_2(c+1)$)** donde c es el índice de la columna.

Algunos ejemplos:

```
[[1, 2, 3], [4, 1, 2]] → Verdadero    fila 1 lo cumple
                          Parte entera(log2 1)=0
                          Parte entera(log2 2)=1
                          Parte entera(log2 3)=1
[[1, 2, 3], [11, 2, 8], [[4, 5, 6]] → Falso ninguna lo cumple
```

d) **IMPORTANTE**: Interpretar cual es la relación entre 2^n y $\log_2(n)$

9) Ejercicio de parcial

Implementar una función usando acumuladores booleanos, que dada una matriz de N x N elementos enteros y un arreglo de N elementos enteros determine si el elemento i del arreglo se encuentra en la fila i de la matriz:

```
public static boolean arregloEnFilas(int[][] mat,
                                      int[] arreglo) ...
```

Casos límites a tener en cuenta:

- *arreglo* está vacío (y la función devuelve verdadero)
- *mat* está vacío (y la función devuelve falso)
- la matriz contiene duplicados (no influye, es suficiente con que estén una vez)

9	2	**4**	5
6	**7**	1	7
3	**5**	9	11
12	8	5	**1**

4
7
5
1

Devuelve True

4	2	13	5
6	7	1	3
3	7	9	11
12	8	5	10

4
7
5
1

Devuelve False

10) Ejercicio de parcial

Implementar un algoritmo que dados una matriz de N x N elementos y un arreglo de N elementos, ambos con elementos enteros >=0, verifique usando acumuladores booleanos que para toda fila i de la matriz se cumpla que sus elementos son múltiplos del elemento i del arreglo, y que alguna columna sea igual al arreglo elemento a elemento.

Casos límites a tener en cuenta:

- *arreglo* está vacío (la función devuelve falso)
- *mat* está vacío (la función devuelve falso)
- la matriz contiene duplicados (no influye, es suficiente con que estén una vez)
- la cantidad de filas de la matriz y el tamaño del arreglo deben ser iguales.

	0	1	2	3
0	4	200	2	24
1	6	33	3	12
2	10	15	5	30

2
3
5

➔ Verdadero
➔ (Col 2 coincide con el arreglo)

	0	1	2	3
0	4	200	2	24
1	6	44	8	12
2	10	5	15	30

2
3
5

➔ Falso