2do Ejercicio tipo parcial

Un dispositivo que lleva un animal bovino en su cuello recolecta datos de un acelerómetro en tres ejes: X Y Z. Cada toma de datos se representa como una secuencia de valores enteros entre 0 y 1023 para cada eje, las secuencias están separadas por -1 (valor no válido para esta lógica). El productor quiere conocer cómo se ha comportado su animal en el transcurso del día y para ello ingresa un patrón de aceleración X Y Z y una cantidad N de repeticiones. Un patrón que se repite una cierta cantidad de veces consecutivas significa que el animal puede estar pastoreando, caminando, rumiando, etc. Dado un valor N y el patrón X Y Z en un arreglo inicializado con -1 (de tamaño igual al arreglo que tiene los datos), hacer un programa en JAVA que:

- Compruebe si el patrón se repitió N o más veces y si es así que elimine del arreglo las secuencias que se siguen repitiendo consecutivamente luego de la cantidad N.

Por ejemplo, si tenemos el siguiente arreglo de datos:

-1	12	22	44	-1	23	34	55	-1	23	34	55	-1	23	34	55	-1	23	34	57	-1

El arreglo patrón X Y Z a analizar es X=23, Y=34, Z=55 y el N=2

-1	-1	-1	-1	-1	-1	-1	-1	23	34	55	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

El arreglo resultante sería

-1	12	22	44	-1	23	34	55	-1	23	34	55	-1	-1	23	34	57	-1	-1	-1	-1

RESOLUCION

```
public class ejercicio2TipoParcial {
    public static final int MAX = 21, SEPARADOR=-1,N=2;
    public static void main(String[] args) {
        int[] arrDatos =
{-1,23,34,55,-1,3,34,55,-1,23,34,55,-1,23,34,55,-1};
        int[] arrPatron =
{-1,-1,-1,-1,-1,-1,-1,-1,23,34,55,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};
        int ini = 0,fin = -1, iniPatron=0, finPatron=-1,cantSecuenciasRepetidas=0;;
        iniPatron=buscarInicio(arrPatron,finPatron+1);
        finPatron=buscarFin(arrPatron,iniPatron);
        while (ini < MAX) {
            ini = buscarInicio(arrDatos, fin + 1);
            if (ini < MAX) {
                fin = buscarFin(arrDatos, ini);
                // esta demas preguntar por el tamaño porque todas son de tamaño 3
                 // se puede dejar igual ya que queda como se venia haciendo
                if ((fin-ini+1==finPatron-iniPatron+1)
&&cumplePatron(arrDatos,ini,fin,arrPatron,iniPatron)){
                    cantSecuenciasRepetidas++;
                    if (cantSecuenciasRepetidas>N){
                        eliminarSecuencia(arrDatos, ini, fin);
                        fin = ini;
                    }
                }
                else
                    cantSecuenciasRepetidas=0;
            }
        }
        mostrarArreglo(arrDatos);
```

```
    }
    public static int buscarInicio(int[] arr, int pos) {
        while (pos <  MAX && arr[pos] == SEPARADOR)
            pos++;
        return pos;
    }
    public static int buscarFin(int[] arr, int pos) {
        while (pos <  MAX && arr[pos] != SEPARADOR)
            pos++;
        return pos - 1;
    }
    public static boolean cumplePatron(int[] arrDatos, int ini, int fin, int[]
arrPatron, int iniP) {
        while (ini <= fin && arrDatos[ini] == arrPatron[iniP]) {
            ini++;
            iniP++;
        }
        return ini > fin;
    }
    public static void eliminarSecuencia(int[] arrDatos, int ini, int fin) {
        for (int i = ini ; i <= fin; i++)
            correrAIzquierda(arrDatos, ini);
    }
    public static void correrAIzquierda(int[] arr, int pos) {
        for (int i = pos; i < MAX - 1; i++)
                arr[i] = arr[i + 1];
    }
    public static void mostrarArreglo(int[] arr) {
        for (int i = 0; i < MAX; i++)
            System.out.print(arr[i] + "|");
        System.out.println();
    }
}
```