**MBSoft - Code Challenge - Pasantías**

Resolver este code challenge preferentemente en lenguaje Java

Los puntos **Bonus** son opcionales y son un plus.

**Sugerimos entregar el presente challenge con un link a un repositorio en github. O en su defecto en un google drive.**

**Ejercicio 1**

Escribir un programa en Java que recorra un arreglo y genere un histograma en base a los números de este. El arreglo se llama myArray y contiene 10 elementos que corresponden a números enteros del 1 al 5. Un histograma representa que tanto un elemento aparece en un conjunto de datos. Por ejemplo, para el arreglo: myArray:=(1,2,1,3,3,1,2,1,5,1) el histograma se vería asi:

1: \*\*\*\*\*

2: \*\*

3: \*\*

4:

5: \*

Nota: No puede importar librerías, solo podrá utilizar comandos básicos de Java. (No puede usar Colecciones, ArrayList, Sets, etc)

**Ejercicio 2**

Se tiene una X en la esquina superior izquierda de un área de 4x4. Se tiene una matriz con 10 elementos. Cada 2 elementos de la matriz corresponden a un movimiento, el primero en el eje horizontal y el segundo en el eje vertical. El numero indica las unidades a moverse y el signo la dirección (positivo para derecha o abajo, negativo para izquierda o arriba)

Por ejemplo, para la matriz myArray:=(1,2,-1,1,0,1,2,-1,-1,-2)

La X se moverá una unidad a la derecha y dos hacia abajo, luego una unidad a la izquierda y una abajo y así sucesivamente. El programa a escribir debe imprimir la posición final de la X. Para representar los lugares donde la X no se encuentra utilizar la letra O. Si la instrucción obliga a la X a salir del área de 4x4 la X permanecerá en el borde, sin salir. Para el arreglo presentado el resultado se vería así:

OXOO

OOOO

OOOO

OOOO

Nota: No puede importar librerías, por lo que el código solo podrá utilizar comandos básicos de Java.

**Ejercicio 3**

Tengo una lista de códigos de producto, el formato de estos códigos es: "XXX-NNNNN-Y"

Donde:

XXX son letras (A-Z) ---> código alfabético del producto

NNNNN son números (0-9) ---> código numérico de la región geográfica

Y es un dígito verificador (0-9)

Se pide:

**a)** Realizar la función:

boolean **esPrioritario**(codigo), donde devuelve true o false si el código de producto comienza con las letras 'P' o 'W'

**b)** Realizar la función:

boolean **verificar**(codigo), donde chequea si el dígito verificador es correcto. Y devuelve true o false si el dígito verificador es correcto.

El dígito verificador se calcula de la siguiente manera: se suma cada cifra del código de la región geográfica => si obtengo un número de más de 2 dígitos, repito (vuelvo a sumar cada dígito) así, hasta llegar a 1 sola cifra, ese será mi dígito verificador.

Ejemplo: prodcuto "DCR-88578-9", se realiza la suma 8+8+5+7+8 = 36. Luego se suma 3+6 = 9. Paro. 9 es mi dígito verificador.

**c)** Realizar una función que recibe una lista de productos (lista/array de los códigos), y las ordena alfabéticamente según su código alfabético (XXX), de menor a mayor.

**d)** Realizar una función que recibe 2 listas de productos (listas/arrays de los códigos) y devuelve una lista/array que representa la Unión de los elementos.

**e)** Realizar una función que recibe 2 listas de productos (listas/arrays de los códigos) y devuelve una lista/array que representa la Intersección de los elementos.