

Prueba Técnica

1. Construya un objeto que reciba un arreglo o una matriz de dimensión N el cual contiene números de tipo entero.

El objeto debe tener los siguientes métodos suponiendo la siguiente definición.

`o = MyMatrix.new(matrix)`

a. `o.dimension` -> Devuelve el número entero que define la dimensión del arreglo o matriz en su mayor profundidad.

b. `o.straight` -> Devuelve true o false según el siguiente criterio: -True: Si el arreglo o matriz contiene la misma cantidad de elementos en cada una de sus dimensiones (Matriz uniforme). -False: Caso contrario.

3. `o.compute` -> Devuelve el número entero resultado de la suma de todos los números incluidos en la matriz sin importar el tamaño o dimensión.

Parámetros para tener en cuenta: -Puede utilizar el lenguaje de programación que desee y con el que se sienta más cómodo. -Considere los siguientes casos como ejemplo. Pasar los casos de ejemplo no necesariamente definirá que el ejercicio está bien construido. Pruebe tantos arreglos o matrices como considere necesario.

Ejemplos: `a = [1,2]` `b = [[1,2],[2,4]]` `c = [[1,2],[2,4],[2,4]]` `d = [[[3,4],[6,5]]]` `e = [[[1, 2, 3]], [[5, 6, 7], [5, 4, 3]], [[3, 5, 6], [4, 8, 3], [2, 3]]]` `f = [[[1, 2, 3], [2, 3, 4]], [[5, 6, 7], [5, 4, 3]], [[3, 5, 6], [4, 8, 3]]]`

```
MyMatrix.new(a).dimension -> 1
MyMatrix.new(b).dimension -> 2
MyMatrix.new(c).dimension -> 2
MyMatrix.new(d).dimension -> 3
MyMatrix.new(e).dimension -> 3
MyMatrix.new(f).dimension -> 3
```

```
MyMatrix.new(a).straight -> true
MyMatrix.new(b).straight -> true
MyMatrix.new(c).straight -> true
MyMatrix.new(d).straight -> true
MyMatrix.new(e).straight -> false
MyMatrix.new(f).straight -> true
```

```
MyMatrix.new(a).compute -> 3
MyMatrix.new(b).compute -> 9
MyMatrix.new(c).compute -> 15
MyMatrix.new(d).compute -> 18
MyMatrix.new(e).compute -> 70
```

```
MyMatrix.new(f).compute -> 66
```

- No existen requerimientos de complejidad, pero entre menos complejidad exista en el cálculo de los métodos mejor se puntuará el ejercicio.
- Los parámetros de calificación son funcionales y de construcción de código. Esfuércese por realizar el ejercicio lo más completo posible. El ejercicio incompleto también suma puntos.
- Asegúrese de que no existen problemas de sintaxis o similares, el código debe poder ser compilado o interpretado dependiendo del lenguaje usado.
- Escriba los comentarios que considere necesarios para explicar los procedimientos o algoritmos descritos.
- El ejercicio deberá quedar subido en un repositorio público de GitHub o GitLab.

2. Construya un objeto que reciba como parámetro un string y reconozca mediante expresiones regulares los siguientes criterios.

El objeto debe tener los siguientes métodos suponiendo la siguiente definición.

```
s = MyArray.new(string)
```

- A. `s.operation` -> Devuelve un booleano según el siguiente criterio -True: Si el texto recibido corresponde a una operación aritmética (+, -, /, *) matemática. -False: Si el texto recibido no corresponde a ninguna operación aritmética matemática o se encuentra mal construida.
- B. `s.compute` -> Devuelve el valor computado de la operación aritmética, se deben considerar los paréntesis '(' ')' como agrupaciones de la operación. Devuelve false en el caso de que la operación no pueda ser computada por paréntesis mal agrupados o en el caso de que `s.operation` sea false.

Parámetros para tener en cuenta: -Puede utilizar el lenguaje de programación que desee y con el que se sienta más cómodo. -Considere los siguientes casos como ejemplo. Pasar los casos de ejemplo no necesariamente definirá que el ejercicio está bien construido. Pruebe tantas cadenas como considere necesario.

Ejemplos: `a = "Hello world"` `b = "2 + 10 / 2 - 20"` `c = "(2 + 10) / 2 - 20"` `d = "(2 + 10 / 2 - 20"`

```
MyArray.new(a).operation -> false
MyArray.new(b).operation -> true
MyArray.new(c).operation -> true
MyArray.new(d).operation -> false
```

```
MyArray.new(a).compute -> false
MyArray.new(b).compute -> -13
MyArray.new(c).compute -> -14
```

```
MyArray.new(d).compute -> false
```

- No existen requerimientos de complejidad, pero entre menos complejidad exista en el cálculo de los métodos mejor se puntuará el ejercicio.
- Los parámetros de calificación son funcionales y de construcción de código. Esfuércese por realizar el ejercicio lo más completo posible. El ejercicio incompleto también suma puntos.
- Asegúrese de que no existen problemas de sintaxis o similares, el código debe poder ser compilado o interpretado dependiendo del lenguaje usado.
- Escriba los comentarios que considere necesarios para explicar los procedimientos o algoritmos descritos.
- El ejercicio deberá quedar subido en un repositorio público de GitHub o GitLab

3. Modele en diagrama entidad relación una base de datos para una peluquería, en la cual se manejarán datos de clientes(nombre, cédula, género, teléfono y correo electrónico), datos de empleados(nombre, cédula, género, fecha de nacimiento, teléfono, fecha de inicio de contrato, fecha final de contrato), roles para los empleados (cajero, peluquero/a), citas entre clientes y peluqueros/as(fecha y hora), pagos de clientes a las citas agendadas (valor, fecha de pago).

Parámetros para tener en cuenta: -Puede modelar usando Enterprise Architecture, Gliffy o cualquier otra herramienta para el modelamiento del diagrama. -El entregable debe ser una imagen en formato PNG o JPG que contenga el diagrama objetivo.

- Considere las tablas relacionales que necesite, tenga en cuenta que entre más normalizada esté la base de datos menos eficiente será. -Comprenda que la base de datos espera ser utilizada en un framework MVC.

Exitos