

1. **Valor (1 punto)**

a) Dado el arreglo let edades = [20, 16, 19, 17, 20, 18, 22, 17] complete donde aparecen las rayas (____) para contar y mostrar cuántas edades dentro del arreglo corresponden a mayores de edad:

```
let edades = [20, 16, 19, 17, 20, 18, 22, 17]
let mayores = 0
```

```
for (let index = 0; index < ____; index++) {
  if ( ____ >= 18){
    ____
  }
}
console.log(mayores)
```

b) Dado el arreglo let datos = [20, 33, 67, 4, 60] complete donde aparecen las rayas (____) para mostrar el promedio de los elementos del arreglo:

```
let datos = [20, 33, 67, 4, 60]
let contador = 0
let suma = 0
```

```
while (contador < ____) {
  suma = suma + ____
  contador++
}
console.log(suma / ____)
```

2. **Valor (1 punto)** Shirley trabaja como programadora en la empresa Data East programando videojuegos. Ella requiere una función que reciba como argumentos los arreglos ["s", "w", "@", "3", "a", "p"] y ["#", "p", "a", "z", "@"] y retorne un arreglo con los elementos que se repiten en ambos, emparejados como una cadena, en este caso: ["@@", "aa", "pp"]. Cree la función que requiere Shirley.

3. **Valor (1 punto)** Mariana, que trabaja en la empresa BioGenetics S.A, requiere una función que al usuario un arreglo con cuatro cadenas de ADN como elementos, cada cadena debe ser de longitud 7. Un ejemplo de un arreglo ingresado por el usuario sería: ["AATGAAC", "GTTTTTC", "GGTAAAT", "CCCCATG"]). Luego de obtener el arreglo del usuario, muestre cuál es la base que más se repite en tales cadenas que forman el arreglo.

Recuerde que una cadena de ADN puede contener las cuatro bases químicas, solo tres bases, solo dos bases o solo una base, por ejemplo:

"AGTTCCAT" cadena formada por 4 bases
"ATTAGATC" cadena formada por 3 bases
"AATTTATA" cadena formada por 2 bases
"CCCCCCC" cadena formada por 1 base

Cree la función que necesita Mariana.

4. **Valor (2 puntos)** Pedro trabaja en una empresa desarrolladora que está haciendo un software para el cliente **orbi** (una empresa multinacional). Pedro requiere una función que reciba como argumento el arreglo de caracteres ["b", "p", "s", "z", "o", "b", "a", "w", "r", "i"] y muestre si es posible formar la palabra **orbi** con los elementos del arreglo.

.En caso de ser posible, quite los elementos del arreglo que no sirven para formar la palabra **orbi** y luego ordene el arreglo para que quede como el siguiente: ["o", "r", "b", "i"].

Cree la función que requiere Pedro. Haga el análisis y luego codifique.