軟體工程師扶持計畫訓練營

Assignment - Week 2

運用任何你學過的 Python、JavaScript 程式語言基本知識，不依賴任何第三方的模組或套件的情況下，完成以下程式要求。

注意：請使用 Python 3 以上的版本進行開發。

要求一：函式與流程控制

完成以下函式，在函式中**使用迴圈**計算最小值到最大值之間，所有整數的總和。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def calculate(min, max):
    # 請用你的程式補完這個函式的區塊
calculate(1, 3) # 你的程式要能夠計算 1+2+3，最後印出 6
calculate(4, 8) # 你的程式要能夠計算 4+5+6+7+8，最後印出 30
```

JavaScript

```
function calculate(min, max){
    // 請用你的程式補完這個函式的區塊
}
calculate(1, 3); // 你的程式要能夠計算 1+2+3，最後印出 6
calculate(4, 8); // 你的程式要能夠計算 4+5+6+7+8，最後印出 30
```

要求二：Python 字典與列表、JavaScript 物件與陣列

完成以下函式，正確計算出員工的平均薪資，請考慮員工數量會變動的情況。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def avg(data):
    # 請用你的程式補完這個函式的區塊
avg({
    "count":3,
    "employees":[
        {
            "name":"John",
            "salary":30000
        },
        {
            "name":"Bob",
            "salary":60000
        },
        {
            "name":"Jenny",
            "salary":50000
        }
    ]
}) # 呼叫 avg 函式
```

JavaScript

```
function avg(data){
	// 請用你的程式補完這個函式的區塊
}
avg({
	"count":3,
	"employees":[
		{
			"name":"John",
			"salary":30000
		},
		{
			"name":"Bob",
			"salary":60000
		},
		{
			"name":"Jenny",
			"salary":50000
		}
	]
}); // 呼叫 avg 函式
```

要求三：演算法

找出至少包含兩筆整數的列表 (Python) 或陣列 (JavaScript) 中，兩兩數字相乘後的最大值。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def maxProduct(nums):
        # 請用你的程式補完這個函式的區塊
maxProduct([5, 20, 2, 6]) # 得到 120 因為 20 和 6 相乘得到最大值
maxProduct([10, -20, 0, 3]) # 得到 30 因為 10 和 3 相乘得到最大值
```

JavaScript

```
function maxProduct(nums){
        // 請用你的程式補完這個函式的區塊
}
maxProduct([5, 20, 2, 6]); // 得到 120 因為 20 和 6 相乘得到最大值
maxProduct([10, -20, 0, 3]); // 得到 30 因為 10 和 3 相乘得到最大值
```

要求四 (請閱讀英文) : 演算法

Given an array of integers, show indices of the two numbers such that they add up to a specific target. You can assume that each input would have exactly one solution, and you can not use the same element twice.

Python

```
def twoSum(nums, target):
        # your code here
result=twoSum([2, 11, 7, 15], 9)
print(result) # show [0, 2] because nums[0]+nums[2] is 9
```

JavaScript

```
function twoSum(nums, target){
        // your code here
}
result=twoSum([2, 11, 7, 15], 9)
console.log(result) // show [0, 2] because nums[0]+nums[2] is 9
```

要求五 (Optional) : 演算法

給定只會包含 0 或 1 兩種數字的列表 (Python) 或陣列 (JavaScript)，計算連續出現 0 的最大長度。

提醒：請勿更動題目中任何已經寫好的程式。

Python

```
def maxZeros(nums):
        # 請用你的程式補完這個函式的區塊
maxZeros([0, 1, 0, 0]) # 得到 2
maxZeros([1, 0, 0, 0, 0, 1, 0, 1, 0, 0]) # 得到 4
maxZeros([1, 1, 1, 1, 1]) # 得到 0
```

JavaScript

```
function maxZeros(nums){
        // 請用你的程式補完這個函式的區塊
}
maxZeros([0, 1, 0, 0]) // 得到 2
maxZeros([1, 0, 0, 0, 0, 1, 0, 1, 0, 0]) // 得到 4
maxZeros([1, 1, 1, 1, 1]) // 得到 0
```