EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

1. Enter 2 lists of numbers in the console:
   Array 1: [2, 9, 7, 6, 7]
   Array 2: [2, 9, 7, 6, 7]
2. Print **EQUAL** if the 2 arrays contain the same elements (same order!) – Print **NOT EQUAL** otherwise

To perform this exercise, you need to code this function.

Function name	isEqual
Parameters	list1 (an array) list2 (an array)
Return value	**Boolean**
Examples	isEqual([1,2,3],[1,2,3]) -> **True**

WARNING: It is NOT allowed to use: list1 == list2: you need to compare each element one by one.

EXAMPLES	
CONSOLE	EXPLANATION
> [1, 2, 3] > [1, 2, 3] >EQUAL	Two arrays are equal.
> [1, 2, 3] > [2, 1, 3] >NOT EQUAL	Two arrays are not equal.
> [1, 2, 3] > [1] >NOT EQUAL	Two arrays are not equal.
>[] >[] > EQUAL	Two arrays are equal.

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO
1. Enter 2 lists of numbers in the console: Array 1: [2, 9, 7, 6, 7] Array 2: [2, 9, 7, 6, 7] 2. Print **EQUAL** if the 2 arrays contain the same elements– Print **NOT EQUAL** otherwise To perform this exercise, you need to code this function.

Function name	isEqual
Parameters	list1 (an array) list2 (an array)
Return value	**Boolean**
Examples	isEqual([1,2,3],[1,3,2]) -> **True**

WARNING: It is NOT allowed to use: list1 == list2: you need to compare each element one by one.

EXAMPLES	
CONSOLE	EXPLANATION
> [1, 2, 3] > [1, 3, 2] >EQUAL	Two arrays are equal.
> [1, 2, 3] > [2, 1, 3] >EQUAL	Two arrays are not equal.
> [1, 2, 3] > [1, 3, 4] >NOT EQUAL	Two arrays are not equal.
>[] >[] > EQUAL	Two arrays are equal.

EXERCICE 3

WHAT YOUR PROGRAM SHALL DO

First you will to implement the following function:

Function name	numberOfCompare
Parameters	array
Return value	The count of number if we found that the previous number is greater than current number
Examples	numberOfCompare ([4,1,3]) → 1

INPUT:

- Enter an array of integers to the console

***array is not given, you have to code it using eval(input())

OUTPUT:

- Print the number of the time a value is greater than the

previous value on the list

EXAMPLES	
CONSOLE	EXPLANATION
> [4,1,3] >1	The answer is 1 because: *1 is NOT greater than 4 (0 found) * 3 is GREATER than 1 (1 found) So, we return 1
> [1, 2,3,5] >3	The answer is 3 because: * 2 is GREATER than 1 (1 found) * 3 is GREATER than 2 (1 found) * 5 is GREATER than 3 (1 found) So, we return 3
> [5,4,3] >3	The answer is 1 because: *4 is NOT greater than 5 (0 found) * 3 is NOT GREATER than 4 (0 found) So, we return 0
> [] >0	Nothing to compare. So, we return 0

EXERCICE 4

WHAT YOUR PROGRAM SHALL DO

First you need to implement the following function:

Function name	sumFromTo
Parameters	An array
Return value	The sum of numbers from start to end values
Examples	sumFromTo ([2, 5]) → 14 Explanation: we start from **2** and we ends at **5** : **2** + 3 + 4 + **5** = 14
Warning	If End value is lower than start value, **you need to return 0**

Then code the main program:

1. The program asks user to enter the start value and the end value:
   Array Of Start and End Value: [2,5]
2. The program will print the sum of numbers between start and end values
   The sum of numbers between 2 and 5 is: 14

Warning: you need to call the function you have defined previously

EXAMPLES	
CONSOLE	EXPLANATION
sumFromTo([3,6]) >18	we start from **3** and we end at **6:** **3** + 4 + 5 + **6** = 18 So, we return 18
sumFromTo([7,6]) >0	End value cannot lower than end value! So, we return 0
sumFromTo([7]) >You got Error!	The user must be given 2 values for this function! So, we return message Error: "You got Error!"

EXERCICE 5

WHAT YOUR PROGRAM SHALL DO

First you need to implement the following function:

Function name	sumBetweenNumberTwo
Parameters	An array
Return value	The sum of numbers between 2
Examples	sumBetweenNumberTwo([1, 2, 5, 4, 5, 6, 12, 2, 3, 4]) → **32** Explanation: we start from **2** and we ends at **2** : 5 + 4 + 5 + 6 + 12 = **32**

Warning: if we have only one number 2 in array, we need to sum until array finished.

EXAMPLES	
CONSOLE	EXPLANATION
sumBetweenNumberTwo([1, 2, 5, 4, 5, 6, 12, 2, 3, 4]) >32	we start from **3** and we end at **6:** 5 + 4 + 5 + 6 + 12 = **32** So, we return 18
sumBetweenNumberTwo([1, 2, 5, 4, 5, 6, 12, 3, 4]) >39	we start from **3** and we end at **6:** 5 + 4 + 5 + 6 + 12 + 3 + 4 = **39** So, we return 18
sumBetweenNumberTwo([1, 5, 4, 5, 6, 12, 3, 4]) >0	We need to check, if there's no number 2 in array, we return 0 So, we return 0
sumBetweenNumberTwo([1, 2, 2, 2, 6, 12, 3, 4]) >25	we start from **3** and we end at **6:** 6 + 12 + 3 + 4 = **25** So, we return 25