MONDAY

EXERCICE 1

WHAT YOUR PROGRAM SHALL DO
- Enter *2 lists* of numbers in the console: **[2, 9, 7, 6, 7]** **[2, 9, 7, 6, 7]** - Print EQUAL if the 2 arrays contains the same elements (same order!) – Print NOT EQUAL otherwise To perform this exercise you need to code this function and call it :

Function name	**isEqual**
Parameters	list1 (an **array**) list2 (an **array**)
Return value	True if list1 and list2 are equal
Examples	**isEqual** ([5, 11], [5, 11]) → True

WARNING:

- **It is NOT allowed to use: list1 == list2 : you need to compare each elements one by one.**

EXAMPLES	
CONSOLE	EXPLANATION
>[1, 2, 3] >[1, 2, 3] >EQUAL	EQUAL
>[1, 2, 3] >[2, 1, 3] >NOT EQUAL	NOT EQUAL
>[1, 2, 3] >[1] >NOT EQUAL	NOT EQUAL
>[1] >[1, 2 , 3] >NOT EQUAL	NOT EQUAL
>[] >[] > EQUAL	EQUAL

```
def isEqual(list1, list2) :
    isEqual = True
    if len(list1) == len(list2) :
        for i in range(len(list1)):
            value1 = list1[i]
```

```
            value2 = list2[i]
            isEqual = isEqual and (value1 == value2)
    else:
        isEqual = False

    return isEqual


# MAIN CODE
list1 = eval(input())
list2 = eval(input())

# Write your code here !
if isEqual(list1, list2) :
    print("EQUAL")
else :
    print("NOT EQUAL")
```

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO

- Enter a text in the console:

Are you ready for algorithm?

- Print the array containing all words of this text:

["Are", "you", "ready", "for", "algorithm?"]

In this program, you will create a function that takes a **string** and return a **list**.
The function must split the string at every space character.

Function name	**splitBySpace**
Parameters	text (a **string**)
Return value	The list of words in the given text (split at every space character)
Examples	**splitBySpace**("Hello first year students") → ["Hello", "first", "year", "students"]

WARNING:

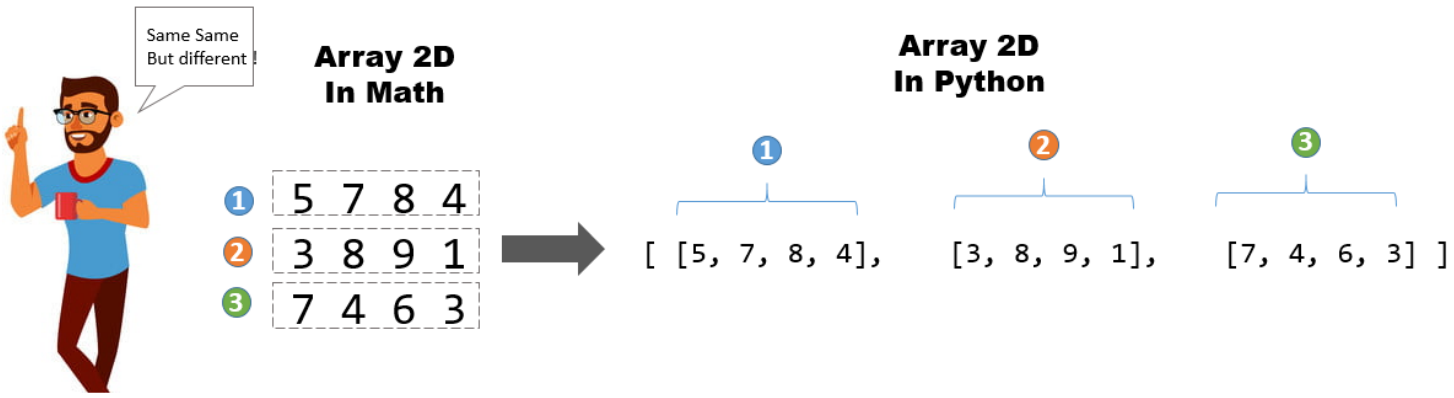
It is NOT allowed to use the PYTHON instruction: split () : you need to code the algorithm by yourself

EXAMPLES	
CONSOLE	EXPLANATION
>hello ronan >['hello', 'ronan']	
>hi >['hi']	
> >[]	If empty string, return an empty list

```
def splitBySpace(theString) :
    words = []

    currentWord =""
    for character in(theString):
        if character == " " :
            if (len(currentWord)>0) :
                words.append(currentWord)    # We add the word only if not
empty
                currentWord = ""             # We reset the current word
        else :
            currentWord +=character

    # We maybe need to add the last word
    if (len(currentWord)>0) :
        words.append(currentWord)

    return words


# MAIN CODE
word = input()

# Write your code here !
print( splitBySpace(word))
```

TUESDAY

How to express an array2D in python?



A array 2D will be represented using an “array of array of numbers”

<u>For instance, how to access to the number 9 in the example above?</u>

- First you access to the second row : `myArray[1]` = > `[3, 8, 9, 1]`
- Then you access to the third element : `myArray[1][3]` = > `9`

WHAT YOUR PROGRAM SHALL DO

- Enter an array 2D in the console:
`[[5, 7, 8, 4], [5, 7, 8, 4], [5, 7, 8, 4]]`
 	
- Replace all 7 numbers by 8 and print the array on console :
`[[5, 8, 8, 4], [5, 8, 8, 4], [5, 8, 8, 4]]`
 	
WARNING:
 	
For this exercise, you cannot create a new array, you need to replace numbers on the SAME array

EXAMPLES	
CONSOLE	EXPLANATION
> [[1, 2, 3], [7, 7, 7]] > [[1, 2, 3], [8, 8, 8]]	We replaced all 7 by 8 in the array 2D
>[] >[]	If empty string, return an empty list
> [[1, 2, 3], [2, 4, 4]] > [[1, 2, 3], [2, 4, 4]]	If no 7 found, nothing to replace !

```
# MAIN CODE
array = eval(input())

# Write your code here !
nbRows = len(array)

for i in range(nbRows):
    nbColumns = len(array[i])

    for j in range(nbColumns):
        value = array[i][j]

        # We replace the value 7 by 8
        if value == 7:
            array[i][j] = 8

print(array)
```

WEDNESDAY

EXERCICE 1

Let's say we have a list of persons: each person has a first name, last name and age:

First Name	Last Name	Age
Sokan	Hy	22
Ronan	Ogor	24

```
Rady          Y             95
Jonathan      Faucher       22
```

In python, we can express this using an array 2D:

[["Lyhor", "Ngorn", 22], ["Ronan", "Ogor", 24], ["Rady", "Y", 95], ["Jonathan", "Faucher", 22]]

WHAT YOUR PROGRAM SHALL DO
- Enter an array 2D in the console *(the persons)* [["Lyhor", "Ngorn", 22], ["Ronan", "Ogor", 24], ["Rady", "Y", 95], ["Jonathan", "Faucher", 22]] - Enter an number in the console *(the* expected *age):* 22 - Print the first name of persons whose age is equal to *expected age:* Lyhor Jonathan - For instance in the example above, we print Sokan and Jonathan because they are all 22 years old.

EXAMPLES	
CONSOLE	EXPLANATION
> [['Bob', 'Y', 19], [['Rady', 'Y', 18]], [['Hugo, 'Panna', 19]] > 19 > Bob > Hugo	Bob and Hugo are 19 years old
> [['Bob', 'Y', 19], [['Rady', 'Y', 18]], [['Hugo, 'Panna', 19]] > 45	Nobody is 45 year old, so nothing printed !

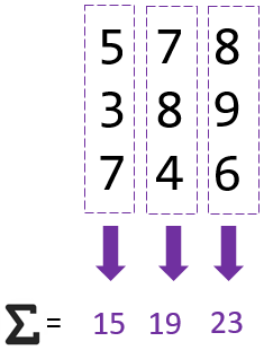
```
# MAIN CODE
persons = eval(input())
age = int(input())

# Write your code here !
for person in (persons):
    if person[2] == age:
        print(person[0])
```

EXERCICE 2

We have an array2D of numbers: for instance:

```
[[5, 7, 8],    [3, 8, 9],    [7, 4, 6]]
```

We want to print the sum of all columns of this array2D:



WHAT YOUR PROGRAM SHALL DO

- Enter an array 2D of numbers in the console :
`[[5, 7, 8],    [3, 8, 9],    [7, 4, 6]]`
 	
- Print the sum of all columns of this array 2D :
`[15, 19, 23]`
 	
- Explanations :
15 = 5 + 3 + 7
19 = 7 + 8 + 4
23 = 8 + 9 + 6

EXAMPLES	
CONSOLE	EXPLANATION
> [[1, 2], [3, 4]] > [4, 6]	First column sum is 4 (1+3) Second column sum is 6 (2 + 4)
> [[5, 7, 8], [3, 8, 9], [7, 4, 6]] > [15, 19, 23]	 15 = 5 + 3 + 7 19 = 7 + 8 + 4 23 = 8 + 9 + 6

```
# MAIN CODE
array = eval(input())

nbRows = len(array)
nbColumns = len(array[0])       # An array 2D : all lines have the SAME

result = []
for columnIndex in range(nbColumns):
    sum = 0

    # we compute the sum of the column at index columnIndex
    for rowIndex in range(nbRows):
```

```
        # We add the value of each row
        sum += array[rowIndex][columnIndex]

    result.append(sum)   # the sum is added to the result

print(result)
```