WEDNESDAY

EXERCICE 0

WHAT YOUR PROGRAM SHALL DO

- Enter a list of names in the console:

[“ronan”, “rady”]

- Enter a new name :

“seiha”

- Print the list with the new name added at the end of the list :

[“ronan”, “rady”, “seiha”]

EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

- Enter a list of numbers in the console:

[2, 9, 7, 6, 7]

- Print the list of numbers which are NOT equal to 7 :

[2, 9, 6]

To perform this exercise you need to code this function and call it :

Function name	**removeSevens**
Parameters	numbers (an **array**)
Return value	the list of numbers NOT equal to 7 (an array)
Examples	**removeSevens** ([5, 7, 7, 11]) ✇ [5, 11]

WARNING:

- You cannot remove numbers 7 from the original array
- you need to create a new ARRAY that contains numbers different from **7**

EXAMPLES	
CONSOLE	EXPLANATION
>[4, 1, 3, 7, 7] >[4, 1, 3]	
>[7, 7, 7] >[]	

CORRECTION

```
def removeSevens(numbers):
    result = []
    for value in numbers:
        if value != 7:
            result.append(value)

    return result

# MAIN CODE
values = eval(input())
print(removeSevens(values))
```

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO

- Enter a list of numbers in the console:

[1, 2, 3, 5]

- We want to add numbers of this list 2 by 2: [1+2, 2+3, 3+5]

So the result would be :

[3, 5, 8]

As you see, the size of the new list is smaller than the original!!

To perform this exercise you need to code this function and call it :

Function name	sum2By2
Parameters	numbers (an **array**)
Return value	A list containing the sum of numbers 2 by 2 (an array)
Examples	sum2By2 ([2,4,5, 1]) ☹ [6, 9, 6]

EXAMPLES	
CONSOLE	EXPLANATION
>[4, 1, 3, 7, 7] >[5, 4, 10, 14]	First we add 4+1 = 5 Then we add 1+3 = 4 Etc.
>[4,5] >[9]	
>[7] >[7]	**If only 1 element, there is no sum, just add the value of this element**
>[] >[]	**Empty list? Just return empty list!**

```
def sum2By2(numbers):
    # Write your code here !
    result = []
    for i in range(1, len(numbers)):
        result.append(numbers[i-1]+numbers[i])
    return result

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

```
# Write your code here !
print(sum2By2(values))
```

THURSDAY

EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

- We enter a list of number in the console :

[10, 5, 6, 10, 7]

- Print “HAS PAIR” if the list contains (at least) 2 numbers with the same value.

- Otherwise print : “HAS NO PAIR”

HAS PAIR

Here : we print HAS PAIR, since we found 2 numbers 10 in this list

EXAMPLES	
CONSOLE	EXPLANATION
>[4, 1, 3, 7, 7] >HAS PAIR	2 numbers 7
>[4, 1, 3, 7, 5] >HAS NO PAIR	Here we haven’t found any pair
>[3, 5, 3, 7, 5] >HAS PAIR	2 numbers 3 and 2 number 5
>[] >HAS NO PAIR	Here we haven’t found any pair

```
alues = eval(input())

hasPair=False
for i in range(len(values)) :
    for j in range(len(values)) :
        if i!=j and values[i] == values[j] :
            hasPair =True

if hasPair:
    print("HAS PAIR")
else:
    print("HAS NO PAIR")# MAIN CODE
v
```

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO

We want to play with cards:
- a card has a value : from 1 to 10
- a card has a color (red “R” or black “B”)

We represent a card using an array of 2 elements :
[<value> , <color>]

Example : *[10 , “R”]   is the card 10 of color RED*

- We enter a list of cards in the console :

[[10, “R”],   [5, “B”] ,   [7, “B”] , [5, “B”]]

- Print “HAS PAIR” if the list of card contains 2 cards with the same value and the same color
    - Otherwise print : “HAS NO PAIR”

HAS PAIR

Here : we print HAS PAIR, since we found 2 cards of the same value + color : **[5, “B]**

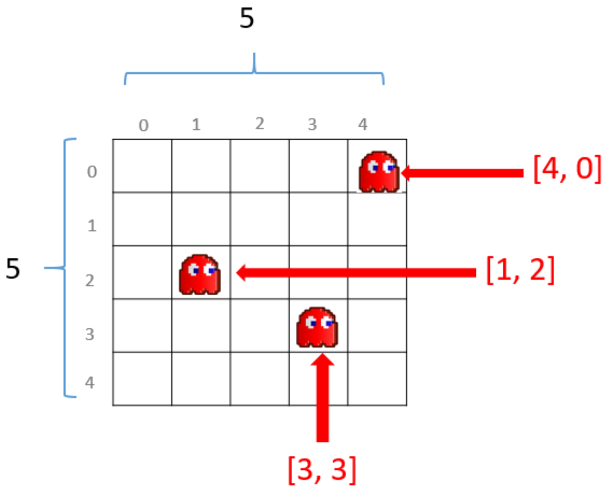
EXAMPLES

CONSOLE	EXPLANATION
>[[4, ‘B’], [4, ‘R’], [3, ‘R’]] >HAS NO PAIR	Here we haven’t found any pair *(we have 2 cards of value 4, but different color)*
>[[4, ‘R’], [3, ‘R’], [4, ‘R’]] >HAS PAIR	2 cards [4, ‘R’]
>[] >HAS NO PAIR	Here we haven’t found any pair

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

hasPair=False
for index1 in range(len(values)) :
    for index2 in range(len(values)) :
        value1 = values[index1][0]
        color1 = values[index1][1]

        value2 = values[index2][0]
        color2 = values[index2][1]

        if index1!=index2 and value1==value2 and color1==color2 :
            hasPair =True

if hasPair:
    print("HAS PAIR")
else:
    print("HAS NO PAIR")
```

FRIDAY

DISPLAY MONSTERS!



EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

<u>We want to display monsters within a grid of 5 X 5 cells:</u>

- a monster has a position on X : from 0 to 4
- a monster has a position on Y : from 0 to 4

We represent a monster position using <u>an array of 2 elements :</u>

[position_X, position_Y]

- Enter a list of monsters position (array of array!)

For instance, this list represent the monsters on above image :

[[3, 3], [1, 2], [4, 0]]

- Print the grid of 5 X5 cells

- Cell with no monster : -
- Cell with monster: *

```
0000*
00000
0*000
000*0
00000
```

To perform this exercise you need to code this function and call it :

Function name	hasMonsterOnCell
Parameters	monsterPositions (**array of array**) : the positions of monsters cellX (integer) : the cell X position cellY (integer) : the cell Y position
Return value	Return True if a monster is on given cell position, given the list of monster position Return False otherwise
Examples	hasMonsterOnCell ([[0, 0], [1, 0]] , 1, 0) ✇ True hasMonsterOnCell ([[0, 0], [1, 0]] , 1, 4) ✇ False

```
def hasMonsterOnCell (monsterPositions, cellX, cellY):
    hasMonster = False
    for monsterPosition in monsterPositions:
        if monsterPosition[0] == cellX and monsterPosition[1] == cellY :
            hasMonster = True
    return hasMonster

# MAIN CODE
allMOnsterPositions = eval(input())

# Write your code here !
result=""
for y in range(5):
    for x in range(5):
        if hasMonsterOnCell(allMOnsterPositions, x, y):
            character = "*"
        else:
            character = "0"

        result+=character
    result+="\n"

print(result)
```