TUESDAY

TUESDAY EXERCICE 1

WHAT YOUR PROGRAMM SHALL DO
- Enter 2 numbers in the console (number1, number2) - Print “WIN” if the number are 4 and 1, otherwise print “LOST”

EXAMPLES	
CONSOLE	EXPLANATION
>Number1 : 4 >Number2 : 1 >WIN	We found the 2 numbers 4 and 1 => we won
>Number1 : 1 >Number2 : 4 >WIN	We found the 2 numbers 4 and 1 => we won
>Number1 : 1 >Number2 : 11 >LOST	We didn’t found the 2 numbers 4 and 1 => we lost

CORRECTION

```
number1 = int(input("Number 1:"))
number2 = int(input("Number 2:"))

if (number1 == 4 and number2 == 1) or (number2 == 4 and number1 == 1):
    print("WIN")
else:
    print("LOOSE")
```

TUESDAY EXERCICE 2

WHAT YOUR PROGRAMM SHALL DO
- Enter a numbers in the console (number) • CONSOLE : Your value: - To be valid, the number must be between 1 and 6 (included) - If not, the program ask to enter again a number, until this one is valid • CONSOLE: Incorrect. Try again: -At the end, print : • CONSOLE: Thanks! Your value <number> is valid.

EXAMPLES	
CONSOLE	EXPLANATION
> Your value: 0 > Incorrect. Try again: 1 > Thanks! Your value 1 is valid	The first input (0) is not in the range [1, 6] The second input (1) is in the range [1, 6] So it’s valid

> Your value: 10 > Incorrect. Try again: 8 > Incorrect. Try again: 6 > Thanks! Your value 6 is valid	The first input (10) is not in the range [1, 6] The second input (8) is not in the range [1, 6] The third input (1) is in the range [1, 6] So it's valid

CORRECTION

```
value = int(input("Your value:"))

while not(value >= 1 and value <= 6):
    value = int(input("Incorrect. Try again:"))

print("Thanks! Your value <number> is valid.")
```

TUESDAY EXERCICE 3

WHAT YOUR PROGRAMM SHALL DO
- Enter 1 number (numberOfValues) in the console • CONSOLE : console shall display : "Number of values:" - Enter < numberOfValues> values in the console - Print the **sum of numbers** which are **before the first number 7** - If there is not number 7, print the sum of all numbers • CONSOLE : console shall display : "Result is: <the result>"

EXAMPLES	
CONSOLE	EXPLANATION
> Number of values: 4 > 4 > 2 > 7 > 1 > Result is : 6	Before the number7, we have the number 4 and2 So the result is : 4 + 2 = 6
> Number of values: 4 > 4 > 2 > 3 > 1 > Result is : 10	Here there is no number 7 So the result is: 4 + 2 +3 + 1 = 10

```
numberOfValue = int(input("Number of value:"))

sum = 0
sevenIsFOund = False
for n in range(numberOfValue):
    value = int(input())

    if value == 7:
        sevenIsFOund = True

    if not sevenIsFOund:
        sum = sum + value

print("Result is:" + str(sum))
```

WEDNESDAY

WEDNESDAY EXERCICE 1

WHAT YOUR PROGRAMM SHALL DO
- Enter 1 number (numberOfValues) in the console CONSOLE : console shall display : "Number of values:" - Enter < numberOfValues> values in the console - Print the sum **of the last 3 numbers** of the list - If the list contains less than 3 numbers, print the sum of all numbers CONSOLE : console shall display : "Result is: <the result>"

EXAMPLES	
CONSOLE	EXPLANATION
> Number of values: 4 > 4 > 2 > 7 > 1 > Result is : 10	The last 3 numbers of this list are : 2, 7, 1 So we print 10 (2 + 7 +1)
> Number of values: 2 > 4 > 2 > Result is : 6	Here we have only 2 values So we print 6 (4 + 2)

COORECTION

```
numberOfValue = int(input("Number of value:"))

sum = 0

for index in range(numberOfValue):
    value = int(input())

    if index >= numberOfValue - 3:
        sum = sum + value

print("Result is:" + str(sum))
```

WEDNESDAY EXERCICE 2

WHAT YOUR PROGRAMM SHALL DO
- Enter a string in the console (word) • CONSOLE : Your word: - This string must contain one character ";" - Example : AA;BBB - Print the characters located before the character ";"

- In this example: AA

- The print the characters located after the character ";"
  - In this example: BBB

- If no character ";" found, print : No semi column found

EXAMPLES	
CONSOLE	EXPLANATION
> Your word: RONAN;HUGO > RONAN > HUGO	The characters before the ";" are : RONAN The characters after the ; are : HUGO
> Your word: RONAN-HUGO > No semi column found	Here we didn't find the character ";"

CORRECTION

```
word = input("Your word:")

indexOfSemiColumn = -1  # -1 means not found

for index in range(len(word)):
    if word[index] == ";":
        indexOfSemiColumn = index

if indexOfSemiColumn == -1:
    print("No semi column found")
else:
    # We split the string into 2 strings
    firstString = word[0:indexOfSemiColumn]
    print(firstString)

    secondString = word[indexOfSemiColumn+1:]
    print(secondString)
```

THURSDAY

THURSDAY EXERCICE 1

WHAT YOUR PROGRAMM SHALL DO
- Enter 1 string (word) in the console CONSOLE : console shall display : "Your word:" - Print the word, with **all "A" replaced by "*"**

EXAMPLES	
CONSOLE	EXPLANATION
> Your word: AARRRA > **RRR*	We have 3 "A" in this word which have bene replaced by a "*"

```
word = input("Your word:")
```

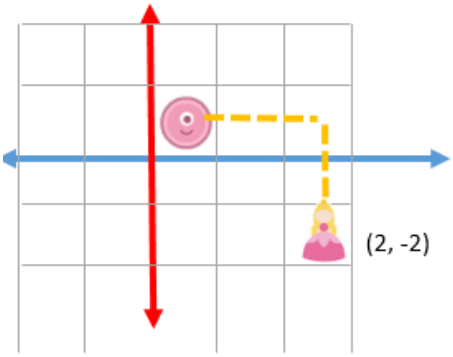
```
result = ""
for index in range(len(word)):
    if word[index] == "A":
        letter = "*"
    else:
        letter = word[index]

    result = result + letter

print(result)
```

FRIDAY

FRIDAY EXERCICE 1

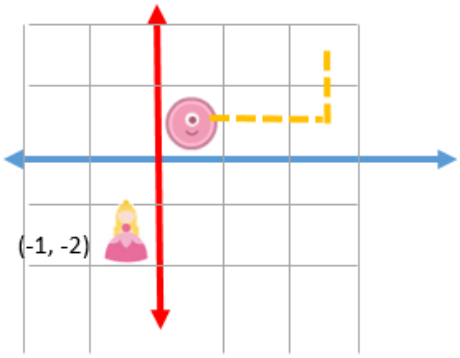
Princess X : 2
Princess Y : -2
Actions : RRDD
>WIN

Because RIGHT + RIGHT + DOWN + DOWN
will bring BALOOK to the princess



Princess X : -1
Princess Y : -2
Actions : RRU
>LOST

Because RIGHT + RIGHT + UP
will not bring BALOOK to the princess



WHAT YOUR PROGRAMM SHALL DO
Enter the **position of the princess** (X first then Y) : • CONSOLE : console shall display : "Princess X : • CONSOLE : console shall display : "Princess Y : Then enter the list of actions : • CONSOLE : console shall display : "Actions: • Enter one string (the actions): which shall contain only: L, R, U, D Regarding each letter of this string, move the position of Balook Print "WIN" if Balook stop on the princess, otherwise print LOST

EXAMPLES	
CONSOLE	EXPLANATION
>Princess X : 2 >Princess Y : -2 >Actions : RRDD >WIN	We move Balook to RIGHT + RIGHT + DOWN + DOWN +DOWN So Balook will be at : (2, -2) Princess is also at : (2, -2) So we WIN
>Princess X : 2 >Princess Y : -3 >Actions : RRDD >LOST	We move Balook to RIGHT + RIGHT + DOWN + DOWN +DOWN So Balook will be at : (2, -2) Princess is at : (2, -3) So we LOST

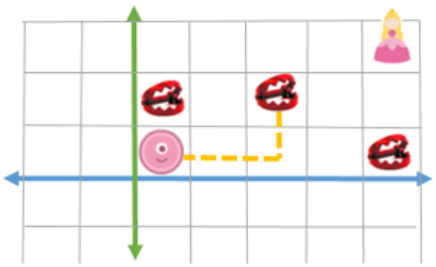
```
actionsString = input("Actions:")
princessX = int(input("Princess X:"))
princessY = int(input("Princess Y:"))

x = 0
```

```
y = 0
for index in range(len(actionsString)):
    action = actionsString[index]

    if action == "R":
        x = x + 1
    elif action == "L":
        x = x - 1
    elif action == "U":
        y = y + 1
    elif action == "D":
        y = y - 1

princessSave = (x == princessX) and (y == princessY)
if princessSave:
    print("WIN")
else:
    print("LOOSE")
```

FRIDAY EXERCICE 2

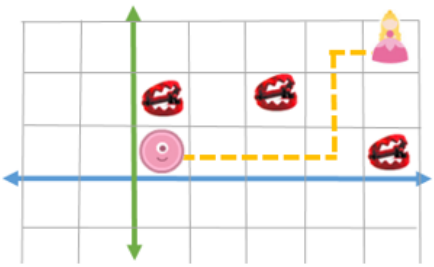
> RRUURRS
> You fell into a trap

Because RIGHT + RIGHT + UP
will make BALOOK fell into a trap
We don't care about the movement after
R R U because we already fall in trap



> RRRUURS
> Princess is saved!

Because RIGHT + RIGHT + RIGHT +
UP + UP + RIGHT will bring BALOOK
to the princess



- Princess is always at position (4, 2)
- Traps are always at positions (0, 1), (2, 1), (4, 0)
- The player wins if the list of move action brings Balook to the princess **without falling in a trap during the travel!!!**

WHAT YOUR PROGRAMM SHALL DO
- Enter one string (the actions): which shall contain only: L, R, U, D - Regarding each letter of this string, move the position of Balook -Print "WIN" if Balook stop on the princess cell (x=4, y=2) without fall in trap on the road -Otherwise print:" LOOSE"

EXAMPLES	
CONSOLE	EXPLANATION
>Actions : LUURRRRR >WIN	We entered 1 LEFT 2 UP and 5 RIGHT - the positions will be: (0;0), (-1;0), (-1;1), (-1;2), (0;2), (1;2), (2;2), (3;2) and (4;2) at the end We never fall in trap and we finish at the princess position, so we print "WIN"
>Actions : RRRRUU >LOOSE	We entered 4 RIGHT and 2 UP - the positions will be: (0;0), (1;0), (2;0), (3;0), (4;0), (4;1) and (4;2) at the end At (4;0) we fall into a trap, so we print "LOOSE"

```
actionsString = input("Actions:")

x = 0
y = 0
balookIsDead = False
for index in range(len(actionsString)):
```

```
    # 1- First move Ballok
    action = actionsString[index]
    if action == "R":
        x = x + 1
    elif action == "L":
        x = x - 1
    elif action == "U":
        y = y + 1
    elif action == "D":
        y = y - 1

    # 2- Check if Ballok is dead : any position among (0, 1), (2, 1), (4, 0)
    if (x == 0 and y == 1) or (x == 2 and y == 1) or (x == 4 and y == 0):
        balookIsDead = True

princessSave = x == 2 and y == 1
if princessSave and not balookIsDead:
    print("WIN")
else:
    print("LOOSE")
```