

# MONDAY

## WHAT YOUR PROGRAM SHALL DO

- **CONSOLE** : console shall display **Enter the 2 numbers :**
- Enter 2 numbers using a string, as follows : **number1;number2**
  - o Number1 and number2 shall be composed of 2 digits (example "45")
  - o Between number1 and number2, the character ";" shall be entered
  - o As example you can enter : "86;23"
- If format is good, display the sum of these 2 numbers
  - o **CONSOLE** : console shall display **Sum is <the sum>**
- Else
  - o **CONSOLE** : console shall display **Format of numbers is wrong**

## EXAMPLES

INPUT	EXPLANATION
>Enter the 2 numbers: 12;13 >Sum is 25	Format is correct So we can print the sum
>Enter the 2 numbers: 12;1 > Format of numbers is wrong	Here the second number contains only 1 digit, not 2
>Enter the 2 numbers: 12/14 > Format of numbers is wrong	Here the separation character is not correct
>Enter the 2 numbers: RONAN > Format of numbers is wrong	Here... everything is wrong!

## HELP ME!!

Here are the steps of the algorithm, can you put them in order?

- Convert string numbers to integer
- Enter the string
- Check the middle character is a ";"
- Do the sum
- Check the second number has 2 digits
- Check the size of the string if correct
- Check the first number has 2 digits

## CORRECTION

```
# 1- Enter position as string
position = str(input("Enter the 2 numbers: "))

# 2- Check the string is composed of 5 characters
has5Character = len(position) == 5

if not has5Character:
    print("Wrong position format")
else:
    # 3- Check we have a number then ; then number
    char0IsNumber = position[0].isnumeric()
    char1IsNumber = position[1].isnumeric()
    char2IsSemiColumn = position[2] == ";"
    char3IsNumber = position[3].isnumeric()
    char4IsNumber = position[4].isnumeric()

    if not (char0IsNumber and char1IsNumber and char2IsSemiColumn and char3Is
Number and char4IsNumber):
        print("Wrong position format")
    else:

        # 4 Convert position into integer and check the range is [0,3]
        value1 = int(position[0:2])
        value2 = int(position[3:5])

        print(value1 + value2)
```

## TUESDAY

### WHAT YOUR PROGRAM SHALL DO

- **CONSOLE** : console shall display **Number** :

Your program shall **continue to ask numbers**, until condition1 or condition2 is checked.

Condition 1: Number 10 has been entered 3 times

Condition 2: The sum of the current number with the previous number is equal to 10

If condition1 or condition2 is checked:

- **CONSOLE** : console shall display **YOU WON**

We don't want the program continue without end : after 6 numbers entered without condition1 or condition2 checked, program stops and:

- **CONSOLE** : console shall display **YOU LOST**

### EXAMPLES

INPUT	EXPLANATION
>Number: 10 >Number: 1 >Number: 10 >Number: 10 >YOU WON	10 has been entered 3 times
>Number: 10 >Number: 14 >Number: 4 >Number: 6 >YOU WON	The sum of current number (6) and previous one (4) is equal to 10
>Number: 1 >Number: 2 >Number: 3 >Number: 4 >Number: 5 >Number: 6 >YOU LOST	6 numbers have been entered, and no condition checked  So the player lost!

### HELP ME!!

Here the big thing is to see how many variables do you need.

Example: Check that number 10 has been entered 3 times": how many variables do you need to this?

## CORRECTION

Tips: you should explain them to go step by step:

- Step1 : only count the number of ten, and no number of tries
- Step2 : add the sum with previous
- Step3 : Manage the number of tries

```
# The 2 conditions to check to stop the loop
numberOfTens = 0
sumWithPrevIs10 = False

# We need to store the previous value
# Note : the first time there is no previous value!
hasPrevValue = False
prevValue = 0

# Number of tries
numberOfTries = 0

# We stop if we found 3 tens OR if the sum of value + previous value is =10
while not (numberOfTens == 3 or sumWithPrevIs10) and numberOfTries < 6:
    value = int(input("Number: "))

    # Update the number of tries
    numberOfTries = numberOfTries + 1

    # Update the number of tens
    if value == 10:
        numberOfTens = numberOfTens + 1

    # Check if the condition sumWithPrevIs10 is checked
    if hasPrevValue and (prevValue + value == 10):
        sumWithPrevIs10 = True

    # Keep the value as prev value
    prevValue = value
    hasPrevValue = True

# End of program : we print the result : WIN or LOOSE
if numberOfTens == 3 or sumWithPrevIs10:
    print("WIN")
else:
    print("LOST")
```

# WEDNESDAY

## EXERCICE 1

### WHAT YOUR PROGRAM SHALL DO

- Enter 4 numbers in the console  
**CONSOLE** : console shall display `Enter number <index> :`
- Display the **min** of the 4 numbers  
**CONSOLE** : console shall display `The min is : <the min>`
- Display the **max** of the 4 numbers  
**CONSOLE** : console shall display `The max is : <the max>`

### EXAMPLES

INPUT	EXPLANATION
>Enter number 1 : 12 >Enter number 2 : 4 >Enter number 3 : -1 >Enter number 4 : 7  >The min is: -1 >The max is: 12	Well it's clear right?

## CORRECTION

Tips: to know we are on the first step, we use the index in the loop

```
min = 0
max = 0
for index in range(4):
    number = int(input("Number "+str(index+1)+"= "))

    if index == 0:
        min = number
        max = number
    else:
        if number > max:
            max = number
        if number < min:
            min = number

print("The minimum is: ", min, "\nThe maximum is: ", max)
```

## EXERCICE 2

### WHAT YOUR PROGRAM SHALL DO

- Enter the number of values:
  - **CONSOLE** : console shall display **Number of values :**
- Enter the < numberOfValues > values :
- If the list of values contains the number 22, display **the sum of even numbers**
- Otherwise , display the **sum of odd numbers**
  - **CONSOLE** : console shall display **Result is: <your result>**

### EXAMPLES

INPUT	EXPLANATION
> Number of values: 4 >1 >10 >2 >3  >Result is : 4	Here, no 22, so we display the sum of odd numbers: $1 + 3 = 4$
> Number of values: 5 >22 >1 >7 >3 >6  >Result is : 28	Here, we have the number 22, so we display the sum of even numbers: $22 + 6 = 28$

```
numberOfValues = int(input("Number of values: "))

sumOfOdd = 0
sumOfEven = 0
contains22 = False
for index in range(numberOfValues):
    value = int(input())

    # Count the number of even numbers
    if (value % 2) == 0:
        sumOfEven = sumOfEven + value
    else:
        sumOfOdd = sumOfOdd + value

    # Check if the value is 22
    if value == 22:
        contains22 = True

# Print the expected result
if contains22:
    print("Result is : " + str(sumOfEven))
```

```
else:  
    print("Result is : " + str(sumOfOdd))
```

## THURSDAY

### *Draw a pyramid of stars!!!*

#### WHAT YOUR PROGRAM SHALL DO

- Enter a number of level:
  - o **CONSOLE** : console shall display **Number of levels :**
- Display a pyramid of "\*" as follows - In this example the pyramid has 4 levels

```
  *  
 ***  
*****  
*****
```

#### EXAMPLES

INPUT	EXPLANATION
> <b>Number of levels: 2</b> > * > ***	Here we have only 2 levels - The first level has 1 star - The second level has 3 stars
> <b>Number of levels: 5</b> > * > *** > ***** > ***** > ***** > *****	

#### HELP ME!!

Here the big thing is to manage the spaces before the stars on each levels

Maybe try to check on each levels:

- How many stars do you have?
- How many space before the stars do you need?

Can you find the way to compute those value during your loop?



## CORRECTION

### Tips:

Before starting the code you should think about the algorithm

Look at the example and for each level count the number of space and the number of stars

Then you will find how to update the number of spaces and star on each iteration

```
numberOfLevels = int(input("Number of levels: "))

numberOfSpaces = numberOfLevels - 1
numberOfStars = 1

result = ""
for i in range(numberOfLevels):

    # Draw the spaces
    for j in range(numberOfSpaces):
        result += " "

    # Draw the stars
    for n in range(numberOfStars):
        result += "*"

    # Update the number of spaces and stars for the next level
    numberOfSpaces = numberOfSpaces-1
    numberOfStars = numberOfStars + 2

    # Carriage return
    result += "\n"

print(result)
```

## FRIDAY

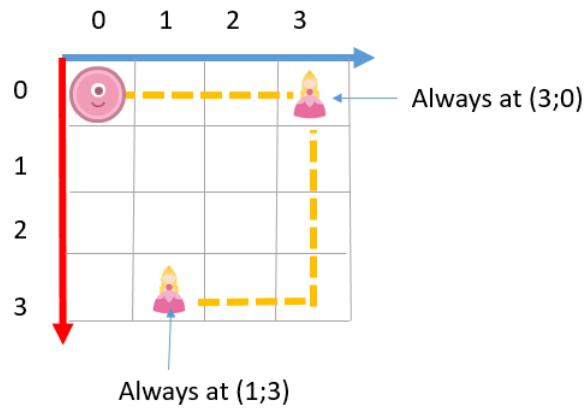
### *Super Balook!*

Now Balook can save many princesses: how ?

- He can go UP, RIGHT, LEFT, DOWN (U, R, L, D)
- He can save a princess with save action (\*)

Example here : **RRR\*DDDLL\***

Balook will save the 2 princesses with those actions.



### Note:

For this exercise, the 2 princess are ALWAYS at given positions: (3;0) and (1;3)

WHAT YOUR PROGRAM SHALL DO
<ul style="list-style-type: none"> <li>- Enter one string (the actions): which shall contain only: L, R, U, D, *</li> <li>- Regarding each letter of this string, move the position of Balook and check if the 2 princess are saved</li> <li>-Print "WIN" if Balook has saved the 2 princesses</li> <li>-Otherwise print:" LOOSE"</li> </ul>

EXAMPLES	
CONSOLE	EXPLANATION
>Actions : RRR*DDDLL* >WIN	We save the 2 princesses
>Action : RDDD*RUUUR*LD >WIN	We save the 2 princesses (we don't need to terminate on any princess position)
>Action : RRR* >LOST	We save only 1 princess

## CORRECTION

Tips: They need ot understand we need 2 boolean, to know if princess1 and 2 were saved

```
# The 2 princess positions
princess1_x = 3
princess1_y = 0

princess2_x = 1
princess2_y = 3

# Get Balook actions
actionsString = input("Actions:")
x = 0
y = 0

princess1Saved = False
princess2Saved = False

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

    elif action == "*":
        if (x == princess1_x and y == princess1_y):
            princess1Saved = True
        if (x == princess2_x and y == princess2_y):
            princess2Saved = True

if princess1Saved and princess2Saved:
    print("WIN")
else:
    print("LOOSE")
```