# **EXERCICE** 1

* Input one word in the console
* Display the number of a and the number of b that the word contains.

***Examples***

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| adada | A:3, B: 0 |
| aaaaa | A:5, B: 0 |
| abbba | A:2, B: 3 |

**Q1** – What will be the **result** for these outputs?

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| aabaa | A:4,B:1 |
| bbccb | A:0, B:3 |
| dcaeb | A:1, B:1 |

**Q2** – **Analyze the symbols** you need to solve this problem.

|  |  |  |
| --- | --- | --- |
| Element | Do you need it? | For what? |
| Action | Yes, I do | Count text of letter “a” and ”b” |
| Decision | Yes, I do | To check condition have letter “a” and “b” or not |
| Repeat | Yes, I do | I want to loop of text |
| Input / Output | Yes, I do | 1, We want to input the text  2, Display the text of a and b |

**Q3** – From the following code, which block is used?

Action

Decision

Input/Output

word = input()

index = 0

nb\_a = 0

nb\_b = 0

while index < len(word):

if word[index] == "a":

nb\_a += 1

if word[index] == "b":

nb\_b += 1

index += 1

print("A:", nb\_a, "B:", nb\_b)

**Q4** – Used this code to create his **flowchart**.

end

end

Print(not the same)

Print(the same)

Count1=count2

Count 2+=1

Len(text2+=”b”)

Len(text2)

Count 1+=1

Len(text1=”a”)

Len(text1)

Get text 2

Get text 1

Count 1= 0

Count 2=0

# **EXERCICE** 2

* Input 2 words in the console
* IF both words contain the same **number of “a”** display THE SAME otherwise display NOT THE SAME

***Examples***

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| dadada  aaa | THE SAME |
| aaaa  aa | NOT THE SAME |
| a  bbbbbbba | THE SAME |

**Q1** – What will be the **results** for those inputs?

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| aa  baa | The same |
| bb  cc | Not the same |
| abba  baab | The same |

**Q2** – Create **flowchart** to solve this problem.

**Q3** – **Implement it**.

# EXERCICE 3

* Input a text in the console.
* Print the number of points related to this word, following the below rules.

|  |  |
| --- | --- |
| **IF THE WORD CONTAINS** | **THEN THE POINTS ARE** |
| **One** ‘*A’* **or more** | 10 points |
| **One ‘***B’* **or more** | 20 points |

* Note: you can cumulate the rules: if you have some “A” and some “B” it will be 10+20 = 30 points!
* If no rules match, then the result is 0 points.

***Examples***

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| KKK | 0 |
| BCA | 30  *Because we found 1 ‘B’ and 1 ‘A’: 10 + 20* |
| MMBBR | 20 |
| MAARTDAC | 10 |
| AABBBB | 30 |
| C | 0 |

**Q1** – What will be the **results** for those inputs?

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| DADADA | 10 |
| ACAAAB | 30 |
| AAAAAA | 10 |
| QWERTY | 0 |

**Q2** – **Fill up the gap** on this flowchart.

Set POINT\_A to 0

Word==”a”

Len(word

Get *WORD*

Yes

Set POINT\_B to 0

No

Done

Set POINT\_A to 10

TEST

If word==”b”

Yes

Set POINT\_B to 20

No

Display POINT\_A + POINT\_B

END

**Q3** – Implement it and test it with the inputs of the first question.

EXERCICE 4

* Execute mentally the below code and write, for each step of execution the value of each variable.
* If the variable is not defined yet, write “?”

a = "roman"

b = a[2]

c = a + b

a = c[-1]

|  |  |  |  |
| --- | --- | --- | --- |
| STEP | A | B | C |
| 1 | roman | ? | ? |
| 2 | Roman | m | ? |
| 3 | Roman | m | romanm |
| 4 | m | m | romanm |