# EXERCISE 1

text = “goodMorning”

**Q1** – Execute these codes.

|  |  |
| --- | --- |
| **Code** | **Result** |
| text[2] | O |
| print( text[-2] ) | N |
| print( text[1:2] ) | o |
| print( text[:4] ) | good |

# EXERCISE 2

* Input text in the console
* Print index of the first letter “O”
  + If no O, write -1

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

|  |  |
| --- | --- |
| **Input** | **Output** |
| goodMorning | -1 |
| Hello | -1 |
| Rady | -1 |

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

|  |  |  |
| --- | --- | --- |
| Element | Do you need it? | For what? |
| Action | Yes, |  |
| Decision | Yes, |  |
| Repeat | Yes, |  |
| Input / Output | Yes, |  |

**Q4**

* Propose **two different ways** to solve this problem
* Create two flowcharts
* Analyze what are the **advantages of each** of your solutions.

# EXERCICE 3

* Input two words in the console
* A number: the sum of uppercases in word1 + word2.

|  |  |
| --- | --- |
| **Input** | **Output** |
| aaBDcc  abcDE | 4  *2 uppercases ("B" and "D") in word1 and 2 updaters (“D” and “E”) in word2* |
| aaZZD  SccDb | 5  *3 uppercases ("Z" and "D") in word1 and 2 updaters (“S” and “D”) in word2* |
| azerty  qwerty | 0 |

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

|  |  |
| --- | --- |
| **Input** | **Output** |
| QZSd  sDFz | 5  *3 uppercases ("Q",”Z” and "S") in word1 and 2 updaters (“D” and “F”) in word2* |
| Asdcv  wxsdE | 2  *1uppercases ( "A") in word1 and 1 updaters ( “E”) in word2* |
| ASDX  wse | 4  *4 uppercases ("A" ,”S”,”D”and "X") in word1* |

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

* For this question, you **cannot** **not** use the **repeat** block
* You can only use the 3 other blocs: Action, Decision, Input / Output

For loop

Count1 = 0

Count2 = 0

sum = 0

upperLetter = True

word1 = input("Enter a word: ")

word2 = input()

for i in range (len(word1)):

if word1[i].isupper() :

Count1 = Count1 + 1

for j in range(len(word2)):

if word2[j].isupper() :

Count2 = Count2 + 1

if sum == 0 and upperLetter== True:

sum = Count1 + Count2

print(sum)

For While

text1=input()

text2=input()

count1=0

count2=0

index=0

result=0

while index < len(text1):

if text1[index].isupper():

count1 += 1

index +=1

index=0

while index <len(text2):

if text2[index].isupper():

count2+=1

index +=1

result =count1+count2

print(result)

# EXERCISE 4

* Input text in the console
* Print index of the LAST letter “K”
* If no K, write -1

**Q1** – Complete the test cases.

|  |  |
| --- | --- |
| **Input** | **Output** |
| DDKDDDKE | 6 |
| DDKDDD | 2 |
| K | 0 |
| AA | -1 |
| xxKK | 3 |
| xKKK | 3 |
| KKKK | 3 |

**Q2** – What is the bug on this code?

text = input()

result = -1

is\_finished = False

for index in range(len(text)) :

    letter = text[index]

    if letter == "K" and is\_finished:

        result = index

      is\_finished = True

print(result)

**Q3** – To find the bug, test this code and check if the output is correct.

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| KK | 1 |
| KKK | 2 |

**Q4** – Correct the code from question 2 to fix the bug.

text = input()

result = -1

is\_finished = False

for index in range(len(text)) :

    letter = text[index]

    if letter == "K" and is\_finished:

        result = index

else:

result=result

     is\_finished = True

print(result)

True

text = input()

indexOfK = -1

for index in range(len(text)) :

    letter = text[index]

    if letter == "K":

        indexOfK = index

print(indexOfK)