# MILLAT UMIDI UNIVERSITY

## COURSE OF COMPUTER SCIENCE
## *LABORATORY PRACTICE n. 9*

### **Exercise 1**:

Write down a Python program which:

- reads a string without spaces from the keyboard (with length at most equal to 50 characters).
- checks whether it is palindrome, displaying a proper message on screen based on the result of the test.

Notice: capital letters should be considered as equivalent to their corresponding small version!

Recall that a string is said to be palindrome when it can be read, equivalently, from left to right and from right to left. For instance, strings `"Anna"`, `"83238"` and `"AbCcBa"` are palindrome.

### **Exercise 2**:

Write down a Python program able to:

- read two words `s1` and `s2`, each one consisting of at most 20 characters.
- generate and display a new string `s3` by removing from `s1` all characters appearing in `s2` (capital letters should be considered as different from the corresponding small ones).

***Example***: The following is a possible execution example (underlined text is typed by the user):

```
Input the s1: Example
Input the s2: exam
Resulting s3: Epl
```

### **Exercise 3**:

Write down a Python program in order to:

- read a single character `ctr`.
- read a sequence of words, each of which with length at most equal to 20 characters. The reading operation terminates when the word `"stop"` is introduced.
- display on screen the word in which `ctr` appears most frequently.

***Example***: The following is a possible program execution (underlined text is typed by the user):

```
Input character: e
Input word: yellow
Input word: green
Input word: orange
Input word: blue
Input word: black
Input word: stop
The word with most 'e' is "green".
```

## **Exercise 4**:

Write a program able to:

- read from the keyboard an unknown number of characters (at most 80), all specified into a single line (terminated by a new line character).
- print on the sub-sequent line the same sequence of characters, where the first character of every "word" has been made uppercase and the remaining ones lowercase.

It is illegal to read and process one word a time: the row introduced by the user must be entirely stored into an array and then properly processed.

***Example***: The following is a possible execution example (underlined text is typed by the user):

```
Input  line: the DEVIL    hides  in the DETAILS.
Output line: The Devil    Hides  In The Details.
```

## **Exercise 5**:

Write down a Python program which:

- reads three strings `s1`, `s2` and `s3` from the keyboard (each one with length at most equal to 50 characters).
- generates a new string by replacing all the occurrences of s2 within s1 with s3.
- outputs such a resulting string.

***Example***: The following is a possible execution example (underlined text is typed by the user):

```
Input 1st string: abcde12345cdefg
Input 2nd string: cde
Input 3rd string: #####
Resulting string: ab#####12345#####fg
```

## **Exercise 6**:

Write a program able to:

- load from the keyboard two pure binary numbers A and B, each one made up of at most 32 bits. The numbers are introduced starting from the most significant bit and expressed using the strictly necessary bits (i.e., only the meaningful ones, skipping the initial 0s).
- compute and display the binary number S given by the sum A+B, with the same format used for the input numbers.

***Example***: The following is a possible execution example (underlined text is typed by the user):

```
Input  A: 1110101
Input  B: 10100
Result S: 10001001
```

as the sum between $A=1110101_2=117_{10}$ and $B=10100_2=20_{10}$ is indeed $S=10001001_2=137_{10}$.