Assignment #5: Functions

Master in Informatics and Computing Engineering
Programming Fundamentals
Instance: 2018/2019

1. Collaborators

List bellow all the people you collaborated with:

up2018xxxxx

up2018xxxxx

. . .

2. SumNumbers

Write a Python function sum_numbers(n) that returns the sum of all positive integers up to and including n.

For example:

- **sum_numbers(10)** returns the value 55 (1+2+3+. . . +10)

3. Perfect

Write a Python function is_perfect(n) to check whether a number n is perfect or not. In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

For example:

- for **n=6** the function returns **True**

- for **n=12** returns **False**

- for **n=28** returns **True**

4. Adigits

Write a Python function adigits that receives 3 strings, each one with a single character, representing a decimal digit. The function should return the highest integer number that you can assemble with the 3 digits given as parameters.

For example:

- **adigits("4", "2", "5")** returns the integer **542**

5. Mastermind

Write a function mastermind to evaluate a single line of a mastermind game. The function receives 6 single character strings. Each string can be "b" for blue, "w" for white and "y" for yellow. The first 3 arguments are the user guess. The last 3 arguments are the correct key. Argument 1 is the user guess for the value at argument 4 and so on. You should give 3 points for each user guess that is completely correct, that is, same color at the same position in the key and 1 point if the user guessed the color right but at the wrong position (that is, the color exists in the key but at another wrong position).

For example:

- **mastermind("b", "w", "y", "b", "w", "y")** returns the integer **9**

6. getPositions

Write a function get_positions that receives two arguments: word_list (a list of strings) and sentence (a string). The second argument contains 2 words that appear in 1st argument concatenated with a single space in between. The function returns a string with the position in the list of the first word and the position of the second word in the list, separated by a single space. Positions start counting at zero.

For example:

- **get_positions(["hello", "world", "lousy"], "lousy world")** returns the string "**2 1**"

The end.

FPRO, 2018/19