

## Written examination – 05/09/2022

Is it your first try?	Yes	No
-----------------------	-----	----

- Section 1: basic questions [max. score: 16]. It contains four simple questions about the topics of the whole course. Each question requires a short answer. Each question answered correctly will give you 4 points (or less for partial answers).
- Section 2: understanding [max. score 8]. It contains an algorithm in Python, and you have to report the particular results of some of its executions according to specific input values.
- Section 3: development [max. score 8] It describes a particular computational problem to solve, and you are asked to write an algorithm in Python for addressing it.

You have 1 hour and 30 minutes for completing the examination. By the final deadline, you should deliver only the original text (i.e. this document) with the definitive answers to the various exercises that must to be written with a pen – pencils are not permitted. You can keep all the draft papers that you may use during the examination for your convenience – blank sheets will be provided to you on request.

## Section 1: basic questions

1 – Which of items in the following lists are true?

- A list is an ordered data structure
- Euler invented the tree data structure
- A dictionary is an ordered data structure
- Assembly is a high-level programming language
- Ada Lovelace written a programme for the Analytical Engine

2 – Consider the following Python function:

```
def f(x, y, z):  
    if x == y:  
        return z[:x]  
    else:  
        return z[:y]
```

What is the value returned by `f(6, 3, "let it snow")`?

3 – Write down a small function in Python that takes in input two strings and returns `True` if they are identical, `False` if they are not identical but contains the same number of characters, otherwise it returns the shorter one.

4 – Introduce the *greedy algorithm* approach and explain which main characteristics a computational problem must have for using a greedy algorithm to solve it successfully.

## Section 2: understanding

Consider the following functions written in Python:

```
def gcs(given_name, mat_string):
    res = 0

    mat_len = len(mat_string)
    for i in range(mat_len // 2):
        sx = mat_string[i]
        dx = mat_string[mat_len - i - 1]

        if sx < dx:
            n = dx - sx
        else:
            n = sx - dx

        res = res + n

    res_s = given_name[res % len(given_name)]
    res_b = res_s in "aeiou"

    return res_s, res_b
```

Consider the variable `my_mat_string` containing the string of all the ten numbers in your matriculation number (e.g. "0000123456"), and the variable `my_given_name` containing string of your given name all in lowercase. What is the value returned by calling the function `gcs` as shown as follows:

```
gcs(my__given_name, my_mat_string)
```

### Section 3: development

A **primality test** is an algorithm for determining whether an input integer greater than zero is prime. The simplest primality test is trial division: given an input number  $n$ , one must check whether it is divisible by any number between 2 and  $\sqrt{n}$  (square root of  $n$ ). If so, then  $n$  is composite; otherwise, it is prime.

Write an algorithm in Python – `def is_prime(n)` – which takes a positive integer  $n$  and returns `True` if  $n$  is prime, otherwise it returns `False`. In the implementation, you must follow strictly the approach introduced above (i.e. checking whether  $n$  is divisible by any number between 2 and  $\sqrt{n}$ ). Hint: to compute the square root of a number in Python, the function `sqrt` of the library `math` can be used – please remember to import it in the code. Some examples of execution:

`is_prime(1)` will return `True`

`is_prime(2)` will return `True`

`is_prime(3)` will return `True`

`is_prime(4)` will return `False`

`is_prime(5)` will return `True`

`is_prime(6)` will return `False`