

## Written examination – 14/03/2022

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 the following sentences are true?

- Python is a high-level programming language
- A natural language is a formal-born language
- Python is a low-level programming language
- Assembly is a low-level programming language
- Assembly is a machine language
- A code in a high-level language is executed directly by the CPU of an electronic computer

2 – Consider the following snippet of Python code:

```
def f(n):  
    result = list()  
    while n > 0:  
        result.append(n)  
        n = n - 1  
    return len(result)
```

Which value is returned by calling the function above as follows:  $f(3)$  ?

3 – Write down a small function in Python that takes in input a string  $s$  and a number  $n$  and returns the character at index  $n$  in the string  $s$  if it exists, otherwise it returns `None`.

4 – Describe, at a general level, the approach adopted in the algorithm presented in the textbook to address the computational problem *find a sequence of moves to solve the peg solitaire*.

## Section 2: understanding

Consider the following functions written in Python:

```
def sc(chars, mat_list):
    n_op = []

    ln = len(mat_list)
    for idx in range(ln // 2):
        cur = mat_list[idx] + mat_list[ln - (1 + idx)]
        n_op.append(cur)

    result = set()
    for n in n_op:
        c = chars[n % len(chars)]
        result.add(c)

    return result
```

Consider the variable `my_mat_list` containing a list of integers where each number is a digit of your matriculation number (e.g. `[0, 0, 0, 0, 1, 2, 3, 4, 5, 6]`), and the variable `my_chars` containing the string with all the alphabetic characters (no spaces!) in lower case included in your full name (given name followed by the family name, i.e. for the name John Doe we have the string `"johndoe"`). What is the value returned by calling the function `sc` as shown as follows:

```
sc(my_chars, my_mat_list)
```

### Section 3: development

An algorithm to compute the **all nearest smaller values** of an input list of numbers **Li** returns a new list of numbers **Lo** where at index **idx** in **Lo** – i.e. **Lo[idx]** – there is the value of the item in the sublist **Li[0:idx]** which is smaller than **Li[idx]** at the index closest to **idx**. For instance, starting from the following list (indexes of the items in the list in red)

0	,	8	,	4	,	12	,	2	,	10	,	6	,	14	,	1
0		1		2		3		4		5		6		7		8

an algorithm computing the all nearest smaller values should return the following list (indexes of the items in the list in red):

-	,	0	,	0	,	4	,	0	,	2	,	2	,	6	,	0
0		1		2		3		4		5		6		7		8

Write an algorithm in Python – `def nearest(list_i)` – which takes in input a list of numbers **list\_i** and returns a new list defining all the nearest smaller values of the items in the input list. The first item of the output list can be set to **None** since it does not have any previous value.