

## Computational Thinking and Programming – A.Y. 2017/2018

Second partial written examination – 11/12/2017

Given name: \_\_\_\_\_

Family name: \_\_\_\_\_

Matriculation number: \_\_\_\_\_

University e-mail: \_\_\_\_\_

Please answer to the following 5 questions [40 minutes max, 1 point each, max score: 5 points]

1. Explain the main differences of how mutable and immutable objects are handled when they are passed to a Python function as input.

2. “Two nodes  $A$  and  $B$  of a tree share the same parent  $C$ ”. If the sentence is true, draw the tree, otherwise, i.e. the sentence is false, provide a justification.

3. Write down the main steps characterising the divide and conquer approach.

4. Consider your matriculation number as a list of numbers stored in the variable `my_list`. What is the result of the execution of the algorithm below when `my_list` is passed as input – i.e. `algorithm(my_list)`?

```
def algorithm(a_list, pos=0):
    if pos >= len(a_list):
        return a_list
    else:
        common_division = pos / 2
        floor_division = pos // 2
        if floor_division < common_division:
            a_list.remove(floor_division)
        else:
            return algorithm(a_list, pos + 1)
```

5. Write the algorithm `def binary_search(item, ordered_list, start, end)`, that takes an item to search (i.e. `item`), an *ordered* list and a starting and ending positions in the list as input, and returns the position of `item` in the list if it is included in it, and *None* otherwise. The approach implemented by the binary search is described as follows. First, it checks if the middle element of the list between `start` and `end` (included) is equal to `item`, and returns its position in this case. Otherwise, 1) if the middle element is lesser than `item` the search is executed in the part of the list that follows the middle element, while 2) if the middle element is greater than `item` the search is executed in the part of the list that precedes the middle element.