

Written examination – 25/01/2019

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 – Select the entities in the following list that are necessary to create a Turing machine:

- an infinite memory tape
- a power plug adapter
- a case for containing the machine
- an head
- building instructions of the machine
- a way to record the current state of the machine, an initial state and zero or more final states
- a table of instructions

2 – Consider the following snippet of Python code:

```
def t(x, y):  
    return x + y - 2  
  
print(t(5, t(3 + 2, 2)))
```

Which value will be printed on screen if we execute such code?

3 – Write down a small function in Python that takes in input a character c and a number k (e.g. "a" and 5) and returns the a string where the input character is repeated k times (e.g. "aaaaa").

4 – Describe the steps of a backtracking algorithm.

Section 2: understanding

Consider the following functions written in Python:

```
def prepare(s):
    l = []

    support = ["9", "8", "7", "6", "5", "4", "3", "2", "1", "0"]

    for idx, c in enumerate(s):
        if c == "0" or c == "9":
            l.append(support[idx])
        else:
            l.append(c)

    return l

def s(l):
    list_r = list(range(len(l)))
    iters = list_r[1:]

    for iter in reversed(iters):
        for idx in range(iter):
            if l[idx] > l[idx + 1]:
                tmp = l[idx]
                l[idx] = l[idx + 1]
                l[idx + 1] = tmp

    return l
```

Consider the variable `my_string_id` containing the **string** of your matriculation number. What is the value returned by calling the function `s` as shown as follows:

```
s(prepare(my_string_id))
```

Section 3: development

The **MostFreqKHashing** is a string hashing technique that takes as input a string s and an integer k and returns the most frequent k characters from the input string s .

Write an algorithm in Python – `def k_hash(s, k)` – which implements the *MostFreqKHashing* hashing technique and returns the set of the most frequent k characters. In case multiple characters have the same frequency, please get the one that occurs first in s .