

## Multimedia Data Modelling

### Exercise

### FINAL EXAM SIMULATION

Write a python program consisting of the following methods:

1. **bitplane**(*im*, *t*, *ptOut*): the method takes as input in image and an integer value *t*. If *t* == 0 then all bit planes of odd position will be set to 0. If *t* == 1 then all even position bit planes will be set to 0. In addition, the method, will save the image in a predefined directory (*ptOut*) with a new name chosen specifically by the user (the operation of putting the name must be provided within the method).
2. **laplacian**(*im*, *ptOut*): the method applies the Laplacian operator calculated by combining appropriate morphological operators on the input image. In addition, the method, will save the image in a predefined directory (*ptOut*) with a new name chosen specifically by the user (the operation of putting the name must be provided within the method).
3. **Random-blocks**(*im*): the method divides the image into non-overlapping blocks of size 8x8 and creates a new image (with the same size as the input) containing all the blocks inserted randomly and not overlapped. The method, will save the image in a predefined directory (*ptOut*) with a new name chosen specifically by the user (the operation of putting the name must be provided within the method).

The python program will have to be structured in order to call one of the previous methods until the letter **q** is inserted (in which case the program terminates its execution). The path **ptIn1**, of the image **image1** and the output path **ptout** where to save the results of some operation must be inserted initially (before any method is called). At each iteration a user can choose which method to call and which image to process (depending on the method to be called).

All methods will return the result of the appropriate operation and overwrite the contents of image1.

### ESSENTIAL ELEMENTS TO RESPECT

- use OPENCV only to read images.
- all methods must be defined via **def**:

```
def name (.....):  
    .....  
    .....  
    return ... # optional
```

- READ THE TEXT CAREFULLY: for example, if a method is expected to return a value, then it must return a value.
- Before writing code, it is strongly recommended to design the program structure on paper and reason about the text!!!

**During the final test, each method implemented is worth a maximum of 10 points.  
Those who solve by the end of the lesson will get the bonus**