CAMBRIDGE INTERNATIONAL UNIVERSITY

COURSE OF COMPUTER SCIENCE
LABORATORY PRACTICE n. 4

Exercise 1 (2 points):

Write down a Python program in order to display, one after the other, the following two series of numbers (where the dots, used for short, must be replaced by the correct numbers).

- First series:

```
(0,0) (0,1) (0,2) (0,3) ... (0,9)
(1,0) (1,1) (1,2) (1,3) ... (1,9)
(2,0) (2,1) (2,2) (2,3) ... (2,9)
...
...
(9,0) (9,1) (9,2) (9,3) ... (9,9)
```

- Second series:

```
 0   1   2   3  ...   9
10  11  12  13  ...  19
20  21  22  23  ...  29
...
...
90  91  92  93  ...  99
```

Exercise 2 (3 points):

Write down a *set* of Python programs (one per figure) able to:

- read a positive integer number *n*.
- display the geometric figures (with "side" equal to *n*) as detailed in the following examples.

Example: assume *n*=4. Then, the figures to be printed are the following ones:

```
****     ****          ****       *+++       *  *
***       ****         *  *       -*++        **
**         ****        *  *       --*+        **
*           ****       ****       ---*       *  *
```

Example: assume *n*=5. Then, the figures to be printed are the following ones:

```
*****     *****         *****      *++++      *   *
****       *****        *   *      -*+++       * *
***         *****       *   *      --*++        *
**           *****      *   *      ---*+       * *
*             *****     *****      ----*      *   *
```

Exercise 3 (2 points):

Write down a Python program able to display a given number of asterisks on a row, repeating this operation as long as the number introduced by the user is positive. In other words, the program must:

- read an integer number *n*.
- if *n* > 0, display *n* asterisks one after the other in one row, then ask for a new value of *n*.

- if $n <= 0$, stop the program execution.

Example: the following is a possible program execution (underlined text is typed by the user).

```
Input n: 5
*****
Input n: 13
*************
Input n: 2
**
Input n: -3
Execution terminated.
```

Exercise 4 (2 points):

Let the Floyd's triangle be defined by the following figure:

```
 1
 2  3
 4  5  6
 7  8  9 10
11 12 13 14 15
 …
```

Write down a Python allowing to:

- read a strictly positive integer number *n*
- display the first *n* rows of the Floyd's triangle (the spacing between numbers for a pretty print is not relevant)

Example: let *n*=3. Then, the program must produce the following output:

```
 1
 2  3
 4  5  6
```

Example: let *n*=4. Then, the program must produce the following output:

```
 1
 2  3
 4  5  6
 7  8  9 10
```

Finally, write another program in order to print only the first *n* numbers of the triangle.

Example: let *n*=5. Then, the program must produce the following output:

```
 1
 2  3
 4  5
```

Example: let *n*=7. Then, the program must produce the following output:

```
 1
 2  3
 4  5  6
 7
```