

## COURSE OF COMPUTER SCIENCE *LABORATORY PRACTICE n. 8*

### **Exercise 1 (1 point):**

Write down a Python program which:

- Reads from the keyboard two positive integer numbers, representing the numerator and the denominator of a fraction.
- Simplifies the fraction as much as possible.
- Prints out the obtained fraction, in the format `num/den`.

**Suggestion:** The fraction can be minimized by dividing the numerator and the denominator by their greatest common divider. Implement a function for finding such a value: the function should have prototype `int gcd(int n1, int n2);`, that is, it receives the two numbers we want to analyze as parameters and returns their greatest common divider as result.

**Example:** The following is a possible execution example (underlined text is introduced by the user):

```
Input numerator   : 12  
Input denominator: 18
```

```
Result: 12/18 = 2/3
```

### **Exercise 2 (2 points):**

Write down a Python function which receives an integer value  $n$  as parameter and returns a “Boolean” value (i.e., an integer assuming only values 1 for representing “true” and 0 for “false”) according to the fact that the number is prime or not. Exploit this function for implementing a program which:

- Reads from the keyboard a positive integer value  $m$ .
- Prints out all the prime numbers smaller than or equal to  $m$ .

**Example:** The following is a possible execution example (underlined text is introduced by the user):

```
Input m: 15  
The prime numbers <= 15 are: 2 3 5 7 11 13
```

### **Exercise 3 (3 points):**

Write down a Python program which, once an integer value  $y$  has been read in from the keyboard, computes and displays the largest integer value  $x$  such that  $x^x < y$ . Solve this problem by implementing, and properly using, a function with prototype `int n_exp_n(int n);`, which computes (and returns) the value  $n^n$  without using the standard `pow` function.

**Example:** The following is a possible execution example (underlined text is introduced by the user):

```
Input y: 650
```

Maximum  $x = 4$

#### **Exercise 4 (3 points):**

Write down a Python program in order to:

- Read an integer value  $n$  from the keyboard.
- Display on screen the graphical representation of an empty array made up of  $N$  elements ( $N$  being a predefined constant), where each cell is shown as a  $n \times n$  square of blanks, as detailed in the following example.

For solving this exercise, implement and properly recall two functions:

1. `void boundary(int n);`  
it displays an entire line like the first (and last) line of the scheme to print.
2. `void internal(int n);`  
it displays an entire line like the other ones appearing in the scheme.

**Example:** let  $N = 5$  and assume that the user introduces value 3 for  $n$ . Then, the following “figure” must be obtained:

```
+---+---+---+---+---+
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
+---+---+---+---+---+
```

Finally, modify the program in order to allow the user to choose the number of elements of the array (instead of considering a constant number of cells, use another variable to store this value: this variable must be then passed to both the functions above).

**Example:** The following is a possible execution example (underlined text is introduced by the user):

```
Input the number of elements   : 8
Input the size of each element: 2
```

```
+---+---+---+---+---+---+---+
| | | | | | | | | |
| | | | | | | | | |
+---+---+---+---+---+---+---+
```