Departamento de Informática

Departamentu de Informática
Department of Computer Sciences

Universidad de Oviedo
Universidá d´Uviéu
University of Oviedo

Degree in Software Engineering – Computing Basics

Unit 2.5 Exercises: Subprograms and Functions

This document includes a collection of exercises from Unit 2.5: Subprograms and Functions. It is recommended that you try doing the exercises without looking at the solutions first, and then you check your answers.

Exercise 1

Write a function that returns the greatest common divisor of two input natural numbers. Assume the function always receives valid input parameters.

Given the above function, write a main program that prints on screen the greatest common divisor of two natural numbers introduced by the user.

Example:

```
Input the first natural number: 42
Input the second natural number: 36
The greatest common divisor is 6
```

Exercise 2

Write a function that returns the factorial `n!` of a non-negative integer number `n` passed as a parameter. Assume the function always receives valid input parameters.

Given the above function, write a main program that prints on screen the factorials of all numbers in the closed interval `[0, k]`, where k is a non-negative integer number specified by the user.

Example:

```
Input the maximum value for which the factorial will be calculated: 5
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
```

Exercise 3

Write a function that returns the sum of all digits in a non-negative integer number passed as a parameter. Then, write another function that returns the number of digits of a non-negative integer number passed as a parameter. Assume the functions always receive valid input parameters.

Given the above functions, write a main program that takes a non-negative integer number from the user, and prints the sequence of numbers that result from the sum of their digits until their corresponding number of digits is 1.

Example:

```
Input a non-negative integer value: 987656789
987656789
65
11
2
```

Exercise 4

Write a function that returns whether a natural number passed as a parameter is prime or not. Think carefully which data type it must return. Assume the function always receives valid input parameters.

Given the above function, write a main program that determines whether the numbers introduced by the user are prime or not. The program must finish when the user introduces a non-natural number.

Example:

```
Input a natural number: 32
32 is NOT a prime number
Input a natural number: 21
21 is NOT a prime number
Input a natural number: 5
5 is a prime number
Input a natural number: -5
```