Duration: 90 minutes.

It is possible to consult:

- any paper material

- the slides downloaded from the course page on the didattica website

- the code of the laboratories, if uploaded to the didattica website in the “elaborati” section

Students caught communicating with each other will be immediately asked to leave the laboratory.

Goldbach’s conjecture states that any even number greater than 2 can be written as the sum of two prime numbers (which can be equal to each other). Goldbach’s conjecture has not been proven theoretically, but is believed to be true on the basis of statistical and probabilistic considerations.

You are asked to create a project with Keil, replace the contents of the startup\_LPC17xx.s file with the one in the **template directory** andadd the other files in the directory. Finally, write **debugged** and **working** subroutines that meet the following 3 specifications.

*Note 1*: You should not change the code inside Reset\_Handler that calls subroutines. It is only required to implement subroutines. It is also recommended not to insert breakpoints in the instructions of Reset\_Handler, but to put them at the beginning of the implemented subroutines.

*Note 2*: Specifications must be completed in order. You can only switch to Specification 2 after verifying that the solution to Specification 1 is working correctly. Same for Specification 3.

**Specification 1 (**5 points). Write in ARM assembly a subroutine isPrimeNumber that receives in input a natural number and returns 1 if the number is prime, 0 otherwise.

The subroutine shall comply with the ARM Architecture Procedure Call Standard (AAPCS) standard, in particular with regard to switching the parameter to input/output and saving the registers.

Example 1: If the input parameter is 12, the return value is 0.

Example 2: If the input parameter is 19, the return value is 1.

**Specification 2** (6 points). Write in ARM assembly a GoldbachConjecture subroutine that receives in input an even natural number. Through a cycle, the subroutine tests all natural number pairs whose sum corresponds to the input number. If both numbers of a pair are first, then Goldbach’s conjecture is verified and the subroutine returns the lesser number in the pair. If there is no pair of prime numbers whose sum is equal to the input number, the subroutine returns 1.

The subroutine shall comply with the AAPCS standard.

Example 1: If the input parameter is 26, the return value is 3 because 26 = 3 + 23.

**Specification 3 (**8 points). Write appropriate instructions / functions in C for:

1. start timer 0 of LPC 1768 board. The timer should not trigger interrupts, but is used exclusively to manage the anti-bouncing of buttons.
2. allow the introduction of a base number 2 by Key1 and Key2 keys. At each push of a button, a bit is added to the right of the number. For example, if the key sequence is Key2, Key2, Key1, Key1, Key2, the final number is 110012 = 25.
3. at the INT0 button press:
   * add a bit to the right of the number previously introduced by the user, to make sure you have an even number. For example, if the user had introduced 110012,after the INT0 pressure the new number is 1100102 = 50.
   * call the subroutine GoldbachConjecture passing the number obtained to the previous point.
   * store in a variable the minimum between 255 and the value returned by the subroutine GoldbachConjecture.
   * display the binary representation of the value of the variable on the led 4-11 (the led 11 corresponds to the bit less significant). For example, 50 = 3 + 47 and so you have to turn on the LEDs 10 and 11.
   * reset the number to allow a new acquisition with the keys Key1 and Key2.