# Exercises

## 11 – Multi - Threading

Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3
56127 Pisa

# Exercise 1

Write a C program with two threads, a producer (P) and a consumer (C). Thread P generates, one at a time, a sequence of numbers by placing them in a single-position buffer shared with thread C. The consuming thread extracts the numbers from the buffer and prints them to `stdout`. If the buffer is full P waits for C to consume the data. Analogously, when the buffer is empty C waits for P to produce a value to consume.

# Exercise 2

## Dining philosophers problem

N philosophers sit at a table with a plate of spaghetti in front of them and a fork on their right and one on their left. To eat spaghetti, a philosopher needs both forks close together. Each philosopher is continuously engaged in a sequence of 3 activities: meditating, trying to acquire forks and eating.

Write a C program that activates N philosopher threads that execute the described loop 100 times. Meditation and the phase where the philosopher eats must be implemented with a variable delay (use for example the **`nanosleep`** system call and the **`rand_r()`** function).

### Hint

To avoid deadlock, define an order for the acquisition of forks by each philosopher.