SDP 2016/17 - Lab 2 - Davide Gallitelli S241521

E03 - Synchronization with semaphores

In order to synchronize threads, the approach taken in this exercise is to define a mutex lock by the means of the *pthread_mutex_t* type, and a *Counter* structure built around it.

```
typedef struct {
    int countA;
    int countB;
    pthread_mutex_t mutex;
} Counter;
```

The mutex manages by itself concurrent modifications on counters *countA* and *countB*, which are used to keep track of the times A and B character have been printed. Moreover, three semaphores are used:

- one for concurrent access to *stdout*, called *print*;
- one to print the second number of each (A or B) type, called *b1*;
- one for iterations, called *b2* .

Very similarly to lab 2.2, the main part of the logic is inside the thread functions, which lock and unlock the mutex in order to access the CR section and print the related characters.