SDP 2016/17 - Lab 2 - Davide Gallitelli S241521

E02 - Synchronization with semaphores

The goal of this exercise is to use semaphores to synchronize several threads. In order to correctly format shared output, two global variables are used to correctly print carriage return when needed, in the form of counters for occurances of character A and B.

Four semaphores are used: - a semaphore to signal all threads to run after the creation process is done

```
/* Signal all threads to run after all of them have been created */
for (int i = 0; i < 3*n; i++)
    sem_post(start);
```

- a semaphore to handle accessing the general printing region, called *print*
- a semaphore for A-type threads, called *printA*
- a semaphore for B-type threads, called *printB*

The thread functions *tfA* and *tfB* are the main brain of the program. Once acquired respectively *printA* or *printB* semaphore, they print the related character on console, and increment the respective counter. The control on counters is done right after that, printing a carriage return only once the counters reach the desired value. If the CR has been printed, 2 more B-type threads and a A-type thread are signaled to run.