

ESERCIZIO ADA

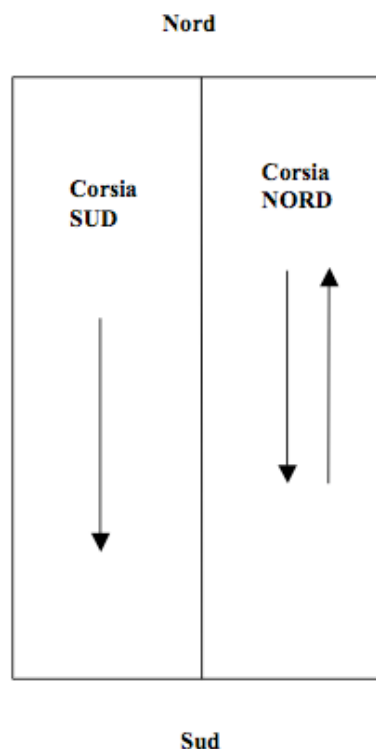
Si consideri un tratto di autostrada, nel quale si stanno svolgendo lavori di manutenzione. Per questo motivo, il tratto considerato e' soggetto a **particolari condizioni di uso**.

In particolare, sono presenti una ***corsia Nord*** e una ***corsia Sud***, ciascuna a capacita' limitata MAX (che esprime il massimo numero di veicoli nella corsia). Nel tratto considerato il traffico e' molto piu' intenso nella direzione NS (Nord->Sud) rispetto alla direzione SN (Sud->Nord).

Pertanto, allo scopo di ridurre i problemi di congestionamento, la gestione del tratto autostradale avviene secondo la seguente politica :

- la corsia Sud viene percorsa a **senso unico nella sola direzione NS**;
- la corsia Nord viene utilizzata a **senso unico alternato** (cioe': non puo' essere contemporaneamente percorsa da veicoli in direzioni opposte) secondo la seguente politica:
 - Quando la corsia Sud e' piena, i veicoli provenienti da nord vengono deviati nella corsia Nord;
 - Nell'accesso alla corsia Nord, si dia la precedenza ai veicoli provenienti da SUD.

Realizzare un'applicazione concorrente in ADA, nella quale il tratto autostradale sia gestito da un task server, che rispetti tutte le specifiche date ed i veicoli siano rappresentati da task clienti.



ESERCIZIO SEMAFORI

Si consideri la **mensa aziendale** di una piccola ditta.

Dall'ora di inizio della "pausa pranzo" in poi, ogni dipendente dell'azienda si reca nel locale della mensa e consuma il proprio pasto, composto (per tutti) da:

- un primo piatto (uguale per tutti)
- un secondo piatto (uguale per tutti).

La mensa è organizzata a "self-service" ed è gestita da un **cuoco** che deposita i piatti su un **bancone** dal quale i dipendenti possono prelevarli.

Si assuma che:

- il numero di dipendenti sia noto a priori e pari alla costante N ;
- il bancone sia di dimensioni limitate pari a MAX , che esprime il numero massimo di piatti che possono essere contemporaneamente depositati su di esso ($MAX < N$).
- ogni dipendente prelevi un piatto alla volta dal bancone;
- ogni dipendente consumi prima il primo e poi il secondo piatto.
- Il cuoco depositi sul bancone un piatto alla volta
- Il cuoco depositi prima tutti i primi piatti;
- prima di depositare i secondi piatti sul bancone, il cuoco attenda che tutti i primi piatti siano stati consumati.

Si realizzi un'applicazione concorrente in **c/linuxthreads** che rappresenti cuoco e dipendenti mediante thread concorrenti e che rispetti i vincoli dati mediante un'opportuna politica di sincronizzazione realizzata tramite semafori.

ESERCIZIO MONITOR

Si consideri un tratto di autostrada, nel quale si stanno svolgendo lavori di manutenzione. Per questo motivo, il tratto considerato e' soggetto a **particolari condizioni di uso**.

In particolare, sono presenti una ***corsia Nord*** e una ***corsia Sud***, ciascuna a capacita' limitata MAX (che esprime il massimo numero di veicoli nella corsia). Nel tratto considerato il traffico e' molto piu' intenso nella direzione NS (Nord->Sud) rispetto alla direzione SN (Sud->Nord).

Pertanto, allo scopo di ridurre i problemi di congestionamento, la gestione del tratto autostradale avviene secondo la seguente politica :

- la corsia Sud viene percorsa a **senso unico nella sola direzione NS**;
- la corsia Nord viene utilizzata a **senso unico alternato** (cioe': non puo' essere contemporaneamente percorsa da veicoli in direzioni opposte) secondo la seguente politica:
 - Quando la corsia Sud e' piena, i veicoli provenienti da nord vengono deviati nella corsia Nord;
 - Nell'accesso alla corsia Nord, si dia la **precedenza ai veicoli provenienti da SUD**.

Si realizzi un'applicazione concorrente in **java** basata su monitor e variabili condizione che rappresenti i veicoli con threads concorrenti e che soddisfi tutti i vincoli dati.

