Nome e cognome _____ N. di matricola (10 cifre num.) _____

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2002/2003
PRIMA PROVA PARZIALE - 12 febbraio 2003

Esercizio -1: essersi iscritti correttamente per svolgere questa prova.
Esercizio 0: Scrivere correttamente nome, cognome e numero di matricola prima di svolgere ogni altro esercizio seguente.

Esercizio 1: Siano dati i seguenti programmi:

```
process P1 {          process P2 {
   UP;                   LEFT;
   DOWN;                 RIGHT;
}                     }
```

Dove UP, LEFT,DOWN e RIGHT sono operazioni atomiche che tracciano un segmento di un'unità rispettivamente verso l'alto, la sinistra, il basso e la destra. Modificando il codice dei due processi usando semafori binari o qualunque costrutto di programmazione sequenziale, fare in modo che l'esecuzione concorrente dei due processi possa produrre un quadrato di lato uno sia tracciato in senso orario sia tracciato in senso antiorario. Nessuna altra figura deve essere possibile. NOTA: scrivere un solo programma formato da due processi, non due programmi separati. Ognuno dei costrutti UP, DOWN, LEFT e RIGHT può comparire una volta sola.

Esercizio 2: I semafori alternati hanno una sola chiamata: S.PV()
Dato un **singolo** processo la prima chiamata è una P, la seconda è una V e così via; tutte le chiamate dispari sono P e quelle pari V. (Nota: l'ordine pari/dispari per le P e V è relativo al singolo processo; in altre parole, se due processi fanno la loro prima chiamata PV ad un semaforo, sarà una P per entrambi). E' possibile assegnare un valore iniziale a scelta del programmatore.
Hanno lo stesso potere espressivo dei semafori generali? Se sì, perchè?

Esercizio 3: In un aeroporto i passeggeri si devono recare prima al check-in poi all'imbarco. La loro vita puo' essere rappresentata dal seguente codice:

```
passenger[i]: process
  .......
  ok=airport.checkin(i,flightid,ticket);
  if (ok) {
    .......
    airport.board(i)
  }
```

L'utente arriva a fare il checkin: puo' avere riservato oppure no. Se l'operazione va a buon fine il passeggero puo' andare all'imbarco.

L'attivita' degli sportelli di checkin puo' essere schematizzata come:

```
checkin_desk[j]: process
  while true do
    noseats=airport.checkinnextflight(j); // Number of seats
    nopass=0; // Number of passengers on the plain
    while (((passenger,ticket)=airport.nextpassenger(j) && nopass<noseats) {
      if reserved(ticket) {
        airport.okpassenger(j,passenger); nopass++;
      } else {
        waitinglist.enqueue(passenger);
      }
    }
    while (!waitinglist.empty() && nopass<noseats) {
      airport.okpassenger(j,waitinglist.dequeue()); nopass++;
    }
    airport.checkinclosed(j)
```

Nella semplificazione dell'esercizio si pensa un solo desk associato ad ogni volo. Quando un desk e' libero chiede un nuovo volo sul quale lavorare. Viene assegnato quello con partenza piu' prossima fra quelli non ancora assegnati. A questo punto i passeggeri fanno il checkin: se avevano prenotato e c'e' posto viene subito confermata l'operazione e data la carta d'imbarco altrimenti vengono posti nella lista d'attesa. Quando il volo viene dichiarato chiuso gli utenti nella lista d'attesa possono occupare i posti ancora liberi sull'aereo.

```
director: process
  while true do
    airport.newflight(flightid,time,capacity);
    airport.closeflight(flightid);
```

Il direttore dello scalo indica l'esistenza di un volo con la chiamata newflight e pochi minuti prima della partenza dichiara il volo chiuso con la chiamata closeflight.