

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2002/2003  
 CONCORRENZA – 30 giugno 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: Passaggio a livello a senso unico alternato**

Una linea ferroviaria a due binari incrocia una strada a due corsie. Ovviamente, sia la sede stradale che quella ferroviaria sono a due corsie. Il passaggio livello, però, è molto stretto, quindi il passaggio per le auto è a senso unico alternato. Per passare, i treni chiamano `void trenoEntra(int direzione)`; una volta passati chiamano `void trenoEsci(int direzione)`; le auto chiamano invece `void autoEntra(int direzione)` e `void autoEsci(int direzione)`. Scrivere il **monitor PassaggioALivello**, avendo cura di evitare starvation, deadlock e attese inutili. Spiegare come il vostro codice evita questi problemi.

**Esercizio 2: Parrucchiera “disonesta”**

Un negozio di parrucchiera è dotato di N sedie per le clienti in attesa, e una sedia per la cliente attualmente servita (una sola parrucchiera, quindi una sola cliente). Quando una cliente arriva, se la parrucchiera è libera, viene subito servita. Se la parrucchiera è occupata, si mette in attesa su una delle sedie; se non c'è più posto, se ne va (`attendiTurno()` ritorna null). La parrucchiera ha alcune clienti “privilegiate”, che quando arrivano passano davanti nella fila (senza far alzare la cliente attualmente servita, però). Le clienti privilegiate sono poche, e quindi aspettano in piedi (impazienti). La vita della parrucchiera e delle clienti normali e privilegiate è la seguente:

**parrucchiera:**

```
while (true) {
  Cliente c = attendiCliente();
  /* taglia i capelli e chiacchera di cose senza senso */
  salutaeringrazia(c);
}
```

**cliente:**

```
Client me = getMyIdentifier();
if (attendiTurno(me)) {
  /* chiacchera di cose senza senso */
  pagaeringrazia(me)
} else
  /* Torna un altro giorno */
```

**cliente-privilegiata:**

```
Client me = < my identifier >
attendiTurnoPrivilegiato(me);
/* chiacchera di cose senza senso */
pagaeringrazia(me);
```

Scrivere le funzioni descritte qui sopra tramite **semafori generali**.