

**UNIVERSITA' DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA**  
**CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2003/2004**  
**COMPITO CONCORRENZA - 2 Settembre 2004**

**Esercizio -1:** essersi iscritti correttamente per svolgere questa prova.

**Esercizio 0:** Su entrambi i fogli, scrivere correttamente nome, cognome, matricola e posizione prima di svolgere ogni altro esercizio.

**Esercizio 1**

Si consideri un seggio elettorale dotato di un presidente e N cabine elettorali (non si considerino gli scrutatori).

La vita di elettori e presidente è la seguente:

<pre>process Presidente {     while (non sono ancora le 22) {         seggio.consegnaScheda();         seggio.consegnaMatita();     } }</pre>	<pre>process Elettore {     seggio.attendiScheda();     int cid = seggio.attendiCabina();     /* cid=0..N-1 è il numero di cabina */     /* Rifletti bene prima di votare... */     /* Riflettici ancora un po'... */     /* Vota! */     seggio.esciCabina(cid); }</pre>
---	---

In altre parole, l'elettore arriva al seggio e attende che il presidente gli consegni una scheda per votare. Ovviamente, nei periodi di poco afflusso può essere il presidente a dover attendere gli elettori. Un volta ottenuta la scheda, l'elettore deve attendere che gli sia consegnata (da parte del presidente) una matita numerata, corrispondente alla cabina. Data la struttura del processo Presidente, in ogni istante vi può essere al più un solo elettore in possesso di scheda ma in attesa di matita.

Ovviamente, le cabine possono essere assegnate ad un solo elettore alla volta. Quando l'elettore esce da una cabina, notifica il presidente (riconsegnando la matita); il presidente può quindi consegnare la matita all' (eventuale) elettore in attesa di matita.

Scrivere il **monitor Seggio**.

**Esercizio 2**

Si consideri il seguente monitor.

```
monitor Counter {
    int i=0;
    condition c;

    procedure entry count() {
        int local = -1;
        i++;
        if (i < 8)
            { local = i; c.wait(); }
        else
            { i = 0; }
        printf("ENTER %d\n", local);
        c.signal();
        printf(" EXIT %d\n", local);
    }
}
```

Si assuma di avere un numero arbitrario di processi che invoca la procedura `entrycount()`; qual è l'output di tale insieme di processi?

**Esercizio 3**

Si consideri la seguente coppia di processi:

<pre>process P {     print T     print R     print A }</pre>	<pre>process Q {     print O     print T }</pre>
--	--

Aggiungere primitive di sincronizzazione basate su semafori tali per cui i due processi stampino solo le parole **torta** o **trota**.