

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2002/2003
 PRIMA PROVA PARZIALE - 19 gennaio 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: Si consideri i seguenti processi che operano sulle variabili condivise a e b, entrambe inizializzate a 0.

```
process P {
    a = a + 2;
    b = a*b;
}
process Q {
    b = b+3;
    a = a*b+7;
}
```

Utilizzando semafori generali inizializzati a 0, modificare il programma precedente in modo tale che gli unici possibili risultati finali siano (a=13, b=3) e (a=19, b=6). Il risultato deve essere dato da un unico programma con due processi, non da due programmi con due processi ciascuno.

Esercizio 2: Si consideri la funzione $\text{minmax}(a,b) = \langle \text{if } (a > b) \{ \text{tmp}=a; a=b; b=tmp; } \rangle$. E' possibile implementare un supporto per le sezioni critiche utilizzando questa funzione? In caso affermativo, illustrare come. In caso negativo, spiegare perchè.

Esercizio 3: Si consideri una rotatoria nel mezzo di una confluenza tra N di strade. Si considerino le seguenti "varianti" del codice stradale:

Variante DX: un'auto ha la precedenza all'ingresso della rotatoria e, una volta nella rotatoria, deve dare la precedenza alle auto che vogliono accedere dai rami intermedi

Variante SX: un'auto deve dare la precedenza all'ingresso della rotatoria e, una volta nella rotatoria, deve ricevere la precedenza dalle auto che vogliono accedere dai rami intermedi.

In entrambi i casi, il verso di rotazione e' antiorario. Ogni arco di rotatoria compreso tra due strade ha una capacità massima CMAX. In altre parole, la capacità totale della rotonda è N*CMAX. Quando un'auto (già nella rotatoria oppure in arrivo da una strada confluyente) vuole entrare in un arco di rotatoria, deve (i) attendere che ci sia capacità residua e (ii) seguire le regole di precedenza. Non è possibile superare all'interno della rotatoria. La "vita" di un auto può essere descritta nel modo seguente:

```
auto_i {
    rotatoria.entra(in);
    for (int j=1; j<=skip; j++)
        rotatoria.saltaUscita((in+j)%N)
    rotatoria.esci((in+skip+1)%N);
}
// Entra dall'entrata in, numerate da 0 a N-1
// Ripeti skip volte; nota che skip può essere uguale a 0
// Salta l'uscita successiva
// Esci dall'uscita prevista
```

3A) Le definizioni delle due varianti possono dare origine a deadlock (entrambe, una delle due o nessuna delle due)? Spiegare. In caso affermativo per entrambe, suggerire un'eventuale variazione delle politiche di precedenza che permetta di evitare deadlock.

3B) Le definizioni delle due varianti possono dare origine a starvation (entrambe, una delle due o nessuna delle due)? Spiegare. In caso affermativo per entrambe, suggerire un'eventuale variazione delle politiche di precedenza che permetta di evitare starvation.

3C) Fornire un'implementazione del monitor rotatoria, che sia esente da deadlock e da starvation.