Nome/cognome _____ N. di matricola (10 cifre) _____ Posizione: Riga _____ Col _____

UNIVERSITA' DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA
CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2004/2005
COMPITO CONCORRENZA – 1 Settembre 2005

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 - Everest

La via sud dell'Everest comprende cinque campi (campo base, campi I II III e IV). I campi sono numerati nel modo seguente: -1 (Kathmandu), 0 (campo base), 1,2,3,4, 5 (cima). Come sapete, durante l'estate, prima della stagione dei monsoni, la via è molto frequentata. Ogni campo **i>0** ha una capacità limitata **c[i]**. Il campo base non ha limite (basta portarsi la tenda). Le autorità del Nepal hanno dovuto decidere un meccanismo di sincronizzazione per le scalate.
L'everest viene gestito da un *monitor* **Everest**, dotato di una **p.e. go_camp(int src, int dest)**, che serve ad uno scalatore a dichiarare che intende lasciare il campo **src** per andare al campo **dest**. . Una volta raggiunta la cima, lo scalatore (tipo Kammerlander) inforca gli sci e se ne torna indietro al campo base. Ogni scalatore opera in modo indipendente, e possono (virtualmente) passare da ogni campo ad ogni altro. La procedura go_camp è bloccante se non c'è posto al campo destinazione o in cima. L'accesso ai campi deve essere gestito in maniera FIFO.

1) Scrivere il monitor Everest
2) Esistono problemi di deadlock? Se sì, quali?
3) Discutere di eventuali problemi di deadlock nel caso la discesa richiedesse una o più soste nei campi intermedi.

Esercizio 2 – Sia dati i seguenti processi:

P1 esegue:

```
function f(i)
{
        if (i<=0)
                V(Y)
        else {
                P(X)
                f(i-1)
                V(X)
        }
}
```

P2 esegue:

```
function g(j)
        if (j<=0)
                V(X)
        else {
                P(Y)
                g(j-1)
                V(Y)
        }
}
```

Se i valori iniziali dei semafori X e Y sono rispettivamente 3 e 4 e ii sia il parametro i sia j valgono 3 iniziallmente:
quale e' l'esecuzione? C'e' deadlock? Quali sono i valori dei semafori alla fine dell'esecuzione o al momento del blocco? Ci sono altre combinazioni di valori che creano deadlock o lo evitano?

Esercizio 3 -

Si vuole realizzare un servizio di spedizione asincrona di messaggi “colorati”: asendc(msg, dest, colore) e arecvc(colore).
asendc non ha valore di ritorno mentre arecvc restituisce il messaggio del colore desiderato (attende se non ve ne sono).
E' possibile specificare il valore “ANY” come parametro di arecv, nel qual caso ogni messaggio puo' essere ricevuto.
Scrivere le funzioni asendc e arecvc facendo uso delle chiamate ordinarie di message passing asincrono asend e arecv.