

## SISTEMI OPERATIVI IDI - Seconda prova intermedia, 14 marzo 2006

### Esercizio 1

Su un sistema sono disponibili 3 classi di risorse (A, B, e C), con 5 istanze ciascuna. In un determinato periodo di tempo operano sul sistema 4 processi (P1, P2, P3, e P4).

Le necessità massime dei processi in termini di risorse sono sintetizzate nella seguente tabella:

	A	B	C
P1	3	2	1
P2	2	2	2
P3	1	1	3
P4	0	1	4

In un certo momento ai diversi processi risultano già allocate alcune risorse, come indicato nella seguente tabella:

	A	B	C
P1	0	1	1
P2	2	2	0
P3	1	0	2
P4	0	1	1

Nelle condizioni descritte sopra

- si determini se il sistema si trova in uno stato sicuro;
- qualora la risposta al punto a) sia affermativa, si indichi una sequenza sicura;
- qualora la risposta al punto a) sia affermativa, si indichi se può essere accolta una richiesta di P4 per un'istanza della risorsa di tipo C;
- sempre nell'ipotesi che la risposta al punto a) sia affermativa, ed in alternativa al punto c), si indichi se può essere accolta una richiesta di P3 per un'istanza della risorsa di tipo B;
- sempre nell'ipotesi che la risposta al punto a) sia affermativa, ed in alternativa ai punti c) e d), si indichi se può essere accolta una richiesta di P1 per un'istanza di una risorsa di tipo B e per un'istanza di una risorsa di tipo C.

Si giustificino adeguatamente le risposte fornite.

## Esercizio 2

Si definisca una classe Java denominata `Riempimento`, in grado di provvedere, in un thread separato, al riempimento un vettore di interi; i valori, generati casualmente, dovranno essere compresi tra  $0$  e  $M$ .

Si definisca inoltre una seconda classe denominata `Complemento` che, dato un vettore di interi, e sempre in un thread separato, provveda alla sostituzione dell'elemento  $i$ -esimo di con il suo complemento a  $M$ ; tra la lettura dell'elemento e la sua sostituzione con il valore complementare deve trascorrere un periodo di tempo casuale (comunque non inferiore a 1 secondo e non superiore a 2 secondi).

Si realizzi infine un programma che, dato un vettore di interi di dimensione  $N$ , e dopo aver creato un numero opportuno di istanze delle classi `Riempimento` e `Complemento`, e avviatane l'esecuzione nei rispettivi threads, provveda anzitutto al riempimento del vettore, quindi alla sostituzione con il complemento di ciascun elemento del vettore, e infine mostri il contenuto finale del vettore.

Nelle due classi e nel programma principale suddetti si gestisca la sincronizzazione tramite la classe `Semaphore` introdotta in Java 5, tenendo conto che il calcolo del complemento deve iniziare solo una volta completato il riempimento, e facendo altresì particolare attenzione al numero di istanze di `Semaphore` da creare (che deve essere minimo) ed alla loro inizializzazione.