

**SISTEMI OPERATIVI IDT/IEL**  
**Seconda prova intermedia, 05 dicembre 2007**

Nome \_\_\_\_\_ Cognome \_\_\_\_\_

Preferenza per l'orale

- 18 dicembre 2007 (appello regolare)  
 **28 dicembre 2007 (data riservata)**  
 **3 o 4 gennaio 2008 (data riservata)**  
 **11 gennaio 2008 (data riservata)**  
 16 gennaio 2008 (appello regolare)

Esercizio 1

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

Le richieste massime dei processi per ciascuna risorsa sono le seguenti:

	<b>A</b>	<b>B</b>	<b>C</b>
<b>P1</b>	2	2	1
<b>P2</b>	2	2	4
<b>P3</b>	1	2	5

Ad un certo istante ai processi risultano già allocate risorse come sotto indicato:

	<b>A</b>	<b>B</b>	<b>C</b>
<b>P1</b>	0	1	1
<b>P2</b>	2	2	0
<b>P3</b>	1	1	2

Nelle condizioni descritte sopra

- a) si determini se il sistema si trova in uno stato sicuro;
- b) qualora la risposta al punto a) sia affermativa, si indichi
  - b.1) una sequenza sicura;
  - b.2) se può essere accolta una richiesta di P3 per un'istanza della risorsa di tipo B;
  - b.3) in alternativa al punto b.2, se può essere accolta una richiesta di P3 per un'istanza della risorsa di tipo C.

Si giustificino adeguatamente le risposte fornite.

## Esercizio 2

Si definisca una classe che, indefinitamente, data una locazione (per locazione si intende un contenitore di dimensione 1), in un flusso di esecuzione dedicato, non appena la locazione si riempie con un dato, lo preleva, quindi lo elabora ed infine deposita il risultato in una pila. Ai fini dello svolgimento dell'esercizio si consideri che i dati trattati siano di tipo stringa e che l'elaborazione consista nel convertire la stringa in una nuova stringa con soli caratteri maiuscoli (si utilizzi a tal fine il metodo `String toUpperCase()` della classe `String`).

Si realizzi quindi un programma che, una volta create  $N$  locazioni  $L_i$  ( $i=1,\dots,N$ ) ed una pila di dimensione  $M$  ( $M>N$ ) avvii altrettanti flussi di esecuzione che realizzano la logica illustrata sopra. Il programma principale crei quindi  $M$  unità di informazione  $D_i$  ( $i=1,\dots,M$ ) e le depositi ciascuna unità in una locazione  $L_i$  secondo uno schema circolare (tipo Round Robin) per poi sospendersi. Quando la pila sarà piena il programma principale riprenda la sua esecuzione svuotando la pila e stampando ciascun elemento da essa estratto.

Per risolvere il problema si utilizzino i semafori di Java.