DIMES UNIVERSITÀ DELLA CALABRIA
Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Esame di **Sistemi Operativi** del Corso di Laurea in Ingegneria Informatica DM270
Prova scritta del 24 aprile 2014 - Durata: 2 ore e 45 minuti - Orario di consegna: _____

Cognome e nome:	Matricola:

Prova 1

Si consideri la seguente successione di riferimenti a pagine in memoria centrale:

1, 2, 3, 4, 1, 5, 6, 1, 3, 2, 3, 7, 6, 3, 2

Calcolare quanti **page fault** si verificano se si usano 3 blocchi di memoria con i seguenti algoritmi di sostituzione:

1. **LRU**
2. **Ottimale**

Prova 2

Si descriva, anche mediante figure opportunamente commentate, la differenza tra allocazione concatenata e indicizzata dei file.

Prova 3

Delle persone si recano in un bar per prendere un caffè. Alcune di queste persone prima pagano alla cassa e poi prendono il caffè al bancone, altre invece prima prendono il caffè al bancone e poi pagano alla cassa. Essendoci una sola cassa, una sola persona alla volta può pagare; il bancone invece è abbastanza largo da poter ospitare quattro persone contemporaneamente. Davanti alla cassa e al bancone vi possono essere delle file. Ciascuna operazione è identificata da un indice intero.

Ogni persona entra nel bar e, se la cassa è libera va prima a pagare e poi a bere il caffè, altrimenti se c'è un posto al bancone libero va prima a bere il caffè e poi a pagare alla cassa, altrimenti controlla la lunghezza delle due file e sceglie di accodarsi alla fila più corta, dando la preferenza alla fila alla cassa nel caso siano di pari dimensione. Una volta completata la prima operazione la persona andrà ad effettuare la seconda, rifacendo eventualmente la fila per la seconda operazione. Ogni persona impiega dai 5 ai 10 secondi per pagare e dai 20 ai 40 secondi per bere il caffè.

Si modelli il sistema descritto in Java, dove le *persone* sono dei thread che interagiscono tramite un oggetto *bar* che espone solo i seguenti metodi:

- **`int scegliEInizia()`**: permette alla persona di cominciare l'operazione di pagamento o di consumazione del caffè e restituisce l'id dell'operazione effettuata
- **`void inizia(int i)`**: permette alla persona di cominciare l'operazione di pagamento o di consumazione del caffè, a seconda del valore dell'indice *i*
- **`void finisci(int i)`**: permette alla persona di terminare l'operazione di pagamento o di consumazione del caffè, a seconda del valore dell'indice *i*

Si implementino due soluzioni che riproducano il funzionamento del problema sopra descritto utilizzando:

- la classe `Semaphore` del package `java.util.concurrent`
- gli strumenti di mutua esclusione e sincronizzazione del package `java.util.concurrent.locks`

Si scriva infine un main d'esempio che, facendo uso di una delle due soluzioni precedenti, inizializzi un oggetto bar, inizializzi 100 persone e ne avvii l'esecuzione.