

Esercizio 1 (10 punti)

In un sistema sono presenti quattro processi P_1, P_2, P_3, P_4 che usano quattro tipi di risorsa A, B, C, D di cui sono presenti rispettivamente 3, 5, 3 e 2 unità. I processi hanno le seguenti matrici di allocazione delle risorse e di massima richiesta di risorse:

	Allocazione				Max			
	A	B	C	D	A	B	C	D
P_1	0	0	0	0	1	1	1	1
P_2	1	1	0	0	2	2	2	2
P_3	1	2	2	0	2	2	2	2
P_4	0	1	1	1	0	1	1	2

Usando l'**algoritmo del banchiere** stabilire se il sistema si trova in uno stato sicuro ed in caso positivo indicare tutte le possibili sequenze sicure. Per tutte le possibili richieste valide (di risorse disponibili e compatibili con la richiesta massima) del processo P_3 determinare se portano in uno stato sicuro o meno.

Esercizio 2 (20 punti)

Cinque contadini producono regolarmente un numero di uova tra 1 e 100 e le depositano in un magazzino centrale con una capienza massima di 2000, il contadino attende se il deposito non può essere fatto. Inoltre in totale il magazzino accetta da ogni contadino un numero massimo di 2000 uova. Se il contadino produce più uova vengono accettate solo quelle fino alla quota massima e raggiunta la quota massima il contadino termina di lavorare. Cinque distributori di uova prelevano regolarmente un numero di uova (tra 1 e 100) dal magazzino, se non ci sono abbastanza uova e se almeno un contadino non ha raggiunto la quota massima devono attendere. Se la richiesta del distributore non può essere soddisfatta (non ci sono abbastanza uova in magazzino e i contadini non possono fornire più uova) il distributore acquisisce le uova disponibili e termina il suo lavoro.

Implementare la situazione descritta in Java usando i metodi **synchronized** per la sincronizzazione tra thread.