

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 una unità per ciascun tipo. 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	1	0	0	1	1	0	0
P_2	0	0	1	0	0	1	1	0
P_3	0	0	0	1	0	0	1	1
P_4	0	0	0	0	1	0	0	1

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) di tutti i processi determinare se portano in uno stato sicuro o meno.

Esercizio 2 (20 punti)

Si vuole realizzare il seguente sistema:

Sono presenti N device che permettono di acquisire il valore di una grandezza fisica, per ogni device un thread *Reader* ogni 200ms legge il valore attuale e lo inserisce in una struttura dati apposita. Un thread *Consumer* deve acquisire gli N valori generati (aspettando se non tutti sono presenti), anche il thread *Reader* deve aspettare se valore generato precedentemente ancora non è stato acquisito dal *Consumer*. Il thread *Consumer* una volta acquisiti gli N valori deve stamparli e calcolarne la media e stamparla.

Per facilitare il testing il device i-esimo deve generare valori casuali in $[i-0.5, i+0.5)$.

Il programma principale deve far partire tutti i thread quindi dopo 10 secondi deve terminare tutti i thread e stampare per ogni thread *Reader* il numero di valori acquisiti dal device.

Realizzare in java il sistema descritto usando i **metodi sincronizzati** per la sincronizzazione tra thread.