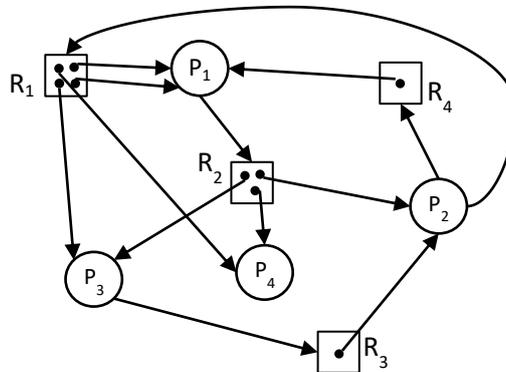


Esercizio 1 (10 punti)

Dato il seguente grafo di allocazione delle risorse:



Costruire la corrispondente tabella di allocazione delle risorse e delle richieste, quindi usando l'algoritmo di rilevazione dello stallo stabilire se il sistema è in stallo o meno. Rivalutare la presenza di uno stallo se il processo P_1 richiede anche una risorsa R_3 .

Esercizio 2 (20 punti)

Si vuole simulare il seguente sistema:

- In un ristorante il menù prevede quattro tipi di piatti
- Ogni piatto prevede l'utilizzo di una quantità (≥ 0) di 10 ingredienti diversi
- La dispensa contiene una quantità massima di ogni ingrediente
- Una richiesta è formata dalle 4 richieste di un certo numero di piatti per ogni tipo, le richieste sono accodate in una coda illimitata
- Sono presenti N cuochi che prelevano una richiesta ma realizzano un solo tipo di piatto, la richiesta rimane in coda fino a quando tutti i tipi di piatti sono assegnati.
- Il cuoco acquisisce gli ingredienti dalla dispensa per i piatti che deve realizzare, potranno essere in numero inferiore se gli ingredienti necessari non sono tutti disponibili, l'acquisizione deve essere fatta in modo atomico.
- Per la preparazione dei piatti assegnati il cuoco deve acquisire un fornello tra i 10 fornelli disponibili.
- Una volta completati tutti i piatti della richiesta (da tutti i cuochi a cui era stata assegnata) questa viene messa in una coda illimitata per la consegna
- Due camerieri attendono sulla coda di consegna e quando prendono la richiesta da consegnare stampano la richiesta fatta e quanti piatti sono consegnati
- Il programma principale deve generare 60 richieste ognuna con un numero casuale di piatti tra 1 e 3 (compresi) dei 4 tipi di piatti, aspettando mezzo secondo tra una richiesta e l'altra.
- Utilizzare i seguenti dati per le quantità degli ingredienti dei 4 tipi di piatti e per le quantità di ingredienti in dispensa
 - ingredienti[0] = new int[]{1,2,2,4,0,0,0,1,0,0}
 - ingredienti[1] = new int[]{0,0,1,1,0,0,2,2,3,1}
 - ingredienti[2] = new int[]{1,0,1,0,1,2,2,2,1,0}
 - ingredienti[3] = new int[]{0,0,0,0,3,0,2,1,0,2}
 - dispensa = new int[]{200,200,400,500,400,200,600,600,400,300}
- Quando tutte le richieste sono state consegnate il programma principale deve terminare tutti i thread

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