

Si vuole implementare in Java il seguente sistema:

- Sono presenti G thread *Generator* che iterativamente generano pacchetti in una coda di ingresso. La coda ha capacità limitata M, se la coda è piena il pacchetto più vecchio viene rimosso.
- I pacchetti sono formati da N numeri interi e da un identificatore progressivo (id)
- L'identificativo del pacchetto è generato tramite un contatore condiviso dai Generator.
- Per facilitare il testing ogni Generator riempie il dati del pacchetto con il numero del Generator.
- Sono presenti K thread *Processor* che prelevano un pacchetto dalla coda di ingresso, calcolano la somma dei valori contenuti e mettono un nuovo pacchetto in una unica coda di uscita. Il pacchetto prodotto ha lo stesso identificatore del pacchetto ricevuto ed il vettore dei dati contiene un elemento con la somma calcolata.
- La coda di uscita (illimitata) deve tenere i pacchetti ordinati per identificatore, in modo da preservare l'ordine di generazione.
- E' presente un thread *Collector* che preleva i pacchetti dalla coda di uscita e iterativamente stampa identificativo e valore della somma.
- Dopo 10 secondi tutti i thread vanno fermati e stampato: il numero totale di pacchetti generati dai Generator, il numero di pacchetti estratti dal Collector, il numero di pacchetti rimossi dalla coda di ingresso quando piena, ed il numero di pacchetti rimasti nelle due code.

Implementare il sistema descritto usando i *semafori* per la sincronizzazione dei thread.