**Traccia dell'Esercizio: Produttori e Consumatori con Semafori**

Implementa un programma Python che simuli la soluzione al problema classico dei produttori e consumatori utilizzando semafori e thread. Il programma deve gestire un buffer condiviso tra produttori e consumatori.

**Requisiti**:

1. Usa un meccanismo di mutua esclusione (mutex) per garantire l'accesso esclusivo al buffer condiviso tra i thread produttori e consumatori.

2. Utilizza semafori per gestire la sincronizzazione tra produttori e consumatori. In particolare, usa due semafori: uno per tenere traccia degli spazi vuoti nel buffer (`empty`) e uno per tracciare gli elementi presenti nel buffer (`full`).

3. Implementa una coda (buffer) con una dimensione massima di `QUEUE\_SIZE`. I consumatori devono prelevare elementi dalla coda e i produttori devono inserire nuovi elementi nella coda.

4. I produttori devono generare un elemento (ad esempio, un numero intero casuale) e inserirlo nel buffer condiviso. I consumatori devono prelevare gli elementi dal buffer.

5. Gestisci un numero specificato di produttori (`N\_PRODUCERS`) e consumatori (`N\_CONSUMERS`), ciascuno eseguito come un thread separato.

Suggerimenti

- Crea una classe `ConsumerThread` che estende la classe `Thread` e implementa il comportamento del consumatore. Utilizza il costruttore per passare i semafori, il mutex e il buffer condiviso.

- Implementa una funzione `produce\_one\_item` per il comportamento del produttore. Questa funzione deve essere eseguita come thread e dovrebbe prendere come argomenti i semafori, il mutex e il buffer condiviso.

- Nel corpo principale (`main()`), crea istanze dei thread consumatori e produttori e avviali. Assicurati di attendere la terminazione di tutti i thread prima di terminare il programma.

Ricorda di gestire correttamente l'accesso concorrente al buffer condiviso, utilizzando semafori e mutex per evitare problemi di sincronizzazione. Buon lavoro!