

[illegible]

## Testo della prova

Si realizzi in linguaggio C/C++ un'applicazione **multithread** basata sul **costrutto Monitor**, per simulare lo streaming video mediante lo schema **produttore-consumatore**. Utilizzare un vettore circolare di buffer condiviso. Ogni buffer è atto a contenere un frame video (una matrice di interi 2x2).

Il monitor dovrà fornire i metodi tradizionali per la consumazione e la produzione di un singolo elemento, con la seguente modifica al metodo di produzione, per consentire al thread di **rallentare le produzioni** in caso di riempimento del vettore. Il prototipo è definito come:

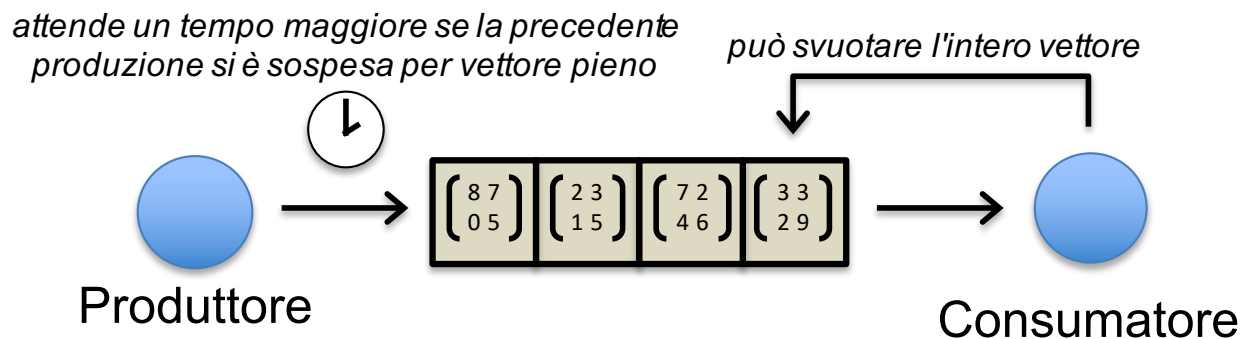
```
int produci(MonitorStreaming *m, Frame f);
```

Al termine della sua esecuzione, il metodo restituisce il valore "0" nel caso in cui la produzione viene effettuata **senza attesa** (ossia, il produttore non trova il vettore pieno, e quindi procede subito nella produzione senza sospendersi su una variabile condition). Il metodo restituisce il valore "1" nel caso in cui la produzione viene invece effettuata **con attesa** (il produttore si è sospeso almeno una volta).

Inoltre, si aggiunga al monitor il seguente metodo, con cui un consumatore può **"svuotare" l'intero vettore di buffer**, ponendo testa=coda=0 e riattivando gli eventuali produttori sospesi:

```
void flush(MonitorStreaming *m);
```

Si simulino un thread produttore e un thread consumatore, con un vettore di 4 buffer. Il produttore effettuerà in totale 12 produzioni in un ciclo, inserendo nei frame valori casuali tra 0 e 10. Ad ogni iterazione, il produttore attende mediante **usleep(tempo\_attesa)**, dove "tempo\_attesa" è una variabile del thread inizialmente posta a 1000000 (1 secondo). Se la produzione subisce una sospensione, il thread dovrà incrementare "tempo\_attesa" di 500000 prima della prossima iterazione. Il consumatore effettuerà in totale 12 consumazioni in un ciclo, attendendo 3 secondi tra le iterazioni.



File da completare:

- main.c
- prodcons.h
- prodcons.c