Esercizio 1: Thread concorrenti

Scrivere un programma in C in cui un thread principale (T) genera 4 thread secondari, di cui 2 del tipo **increment_thread** e 2 del tipo **decrement_thread**.

Tutti i thread condividono una sola variabile intera, inizializzata a 0.

Ogni increment_thread, ciclicamente, incrementa di un'unità la variabile condivisa se e solo se il valore della stessa è inferiore a 10.

Ogni decrement_thread, ciclicamente, decrementa di un'unità la variabile condivisa se e solo se il valore della stessa è superiore a 5.

L'iterazione termina dopo 7 cicli, indipendentemente dal fatto che l'operazione di incremento (o decremento) sia stata svolta o meno in ognuno di essi.

Solo quando la variabile condivisa viene modificata, un thread stampa a video:

- L'operazione eseguita.
- Il valore della variabile condivisa, prima e dopo l'operazione.

Tutti i thread ad ogni iterazione vanno in sleep per un secondo (chiamare la funzione sleep fuori dalla sezione critica).

Il thread principale dopo aver creato i thread secondari attende la terminazione di tutti i thread.

Utilizzare il meccanismo dei pthread_mutex per regolare l'accesso in sezione critica.

Esercizio 2: Simulatore di battaglie Risiko

Scrivere un programma in C in cui un processo padre (P) genera due processi figli P1 e P2.

Il processo padre comunica con i processi figli tramite due separate code di messaggi (1 per P1 e 1 per P2).

Il processo padre ciclicamente, per 10 volte, esegue le seguenti operazioni:

1. Riceve un messaggio da ogni figlio contenente tre interi ordinati in modo decrescente.
2. Confronta il primo intero di P1 (ossia, quello con valore maggiore tra i tre interi di P1) col primo intero di P2 (ossia, quello con valore maggiore tra i tre di P2). Il processo che ha inviato l'intero maggiore guadagna 1 punto.
3. Confronta il secondo intero di P1 col secondo intero di P2. Il processo che ha inviato l'intero maggiore guadagna un punto.
4. Confronta il terzo intero di P1 col terzo intero di P2. Il processo che ha inviato l'intero maggiore guadagna un punto.
5. Invia ad ogni processo figlio un messaggio con l'indicazione dei punti totalizzati.

Dopo 10 cicli il processo padre attende la terminazione dei figli e termina.

Ogni processo figlio ciclicamente, per 10 volte, esegue le seguenti operazioni:

1. Calcola tre numeri interi random compresi tra 1 e 6.
2. Li invia in un messaggio ordinati in modo decrescente.
3. Riceve dal padre il punteggio totalizzato e lo stampa a video.

Dopo 10 cicli i processi figli terminano.

P1	P2	Punti
6	5	+1 a P1
4	5	+1 a P2
2	1	+1 a P1