

### Esercizio 3. Thread

Scrivere un programma in C che simuli un incontro di “tiro alla fune” tra due 2 thread “giocatori” `tp[0]`, `tp[1]`.

E’ data una variabile globale intera `posizione` (con valore iniziale 0) condivisa da tutti i thread. Sono date inoltre due variabili globali intere, `vittorie_tp0` e `vittorie_tp1`.

Ogni thread giocatore esegue un ciclo in cui:

- genera un intero casuale `recupero` compreso tra 0 e 3
- genera un intero casuale `forza` compreso tra 0 e 5
- attende `recupero` secondi
- se `tp[0]`:
  - se `posizione >= 10` riconosce la vittoria di `tp[1]` e:
    - incrementa `vittorie_tp1`
    - setta `posizione = 0`
    - sveglia `tp[1]`
  - altrimenti:
    - decrementa `posizione` di `forza`
    - se `posizione <= -10` ha vinto, e si mette in attesa di `tp[1]`
- se `tp[1]`:
  - se `posizione <= -10` riconosce la vittoria di `tp[0]` e:
    - incrementa `vittorie_tp0`
    - setta `posizione = 0`
    - sveglia `tp[0]`
  - altrimenti:
    - incrementa `posizione` di `forza`
    - se `posizione >= 10` ha vinto, e si mette in attesa di `tp[0]`

(Opzionale) quando uno dei giocatori ha raggiunto 10 vittorie interrompere il gioco, entrambi i giocatori `tp[0]`, `tp[1]` devono aver terminato la loro esecuzione, e la funzione `main()` se ne deve accorgere scrivendo sullo standard output il giocatore che ha totalizzato più vittorie.

**Tempo a disposizione: 50 minuti.**