

<https://tinyurl.com/threadB5dic>

Thread (sia in C che in Java) -- Tempo a disposizione: 50 minuti

N.B.: Consegnare tutti i file sorgente prodotti (C e Java).

Scrivere in C e in Java un programma con un thread *T1* e *T2* che condividono una variabile *m* intera, che va inizializzata a 0.

I thread eseguono un ciclo infinito, comportandosi entrambi come segue:

- scrive *m* sulla standard output e ne salva il valore
- genera un valore casuale intero e lo memorizza in *m*
- attende 300 ms e successivamente controlla il precedente valore di *m*, sia *M0*
 - se *M0* è dispari, il thread *T1* si mette in attesa (e lo scrive prima)
 - se *M0* è pari, il thread *T2* si mette in attesa (e lo scrive prima)

Un terzo thread, *T3* esegue un ciclo infinito in cui sveglia *T1* e *T2*, aspetta 2 secondi e ricomincia.

N.B.

- in C, la chiamata `usleep(ut)` attende per *ut* microsecondi; in Java il metodo `sleep(mt)` della classe `Thread` attende per *mt* millisecondi.
- In C, ciascun thread esegue una funzione diversa `fun_1()` per *T1*, `fun_2()` per *T2* e `fun_3()` per *T3*.
in Java, ciascun thread è istanza di una classe diversa, `Thread1` per *T1*, `Thread2` per *T2* e `Thread3` per *T3*.

Archiviare i file sorgente in un file denominato `nome_cognome_matricola.zip` e caricarli entro 50 minuti all'indirizzo:

<https://tinyurl.com/y73szemv>

Al termine dei 60 minuti non sarà più possibile caricare ulteriori files.