

Una variabile `int n`, inizializzata a 100, è condivisa tra 2 thread `tI`, `tD`.

Il thread `tI`, ciclicamente:

1. attende 100 ms (N.B.: la chiamata `usleep(t)` attende per `t` **microsecondi**)
2. genera un `int` casuale tra 0 e 9 e lo **somma** alla variabile condivisa `n`
3. se `n` è maggiore di 150 termina
4. altrimenti ricomincia da (1), a meno che abbia già compiuto 1000 iterazioni, nel qual caso termina.

Il thread `tD`, ciclicamente:

1. attende 300 ms (N.B.: la chiamata `usleep(n)` attende per `n` **microsecondi**)
2. genera un `int` casuale tra 0 e 9 e lo **sottrae** dalla variabile condivisa `n`
3. se `n` è minore di 80 termina
4. altrimenti ricomincia da (1), a meno che abbia già compiuto 1000 iterazioni, nel qual caso termina.

(Non ricorrere a un array di 2 thread per l'implementazione!)

Il programma termina quando tutti i thread hanno terminato la propria esecuzione. I thread scriveranno di essere terminati. Possono anche visualizzare, a ogni ciclo, il valore trovato in `n`.

Nel codice, proteggere opportunamente la variabile `n` dagli accessi concorrenti.

Tempo a disposizione: 30 minuti.