

1. Socket (C o Java)

Scrivere un server che si metta in ascolto sul **port 3333** in grado di rispondere ad un messaggio composto da una sola stringa `str` composta **da cifre numeriche (da 0 a 9)** terminata dal carattere `\n`.

Il server:

- Converte la stringa ricevuta `str` in un numero intero `n`.
- Il numero viene usato come input per il metodo `"int cubo(int n)"` che restituisce il cubo dell'intero `n`.
- Il server invia al client il valore restituito dal metodo `cubo()`.

Testare il server con un semplice client o con telnet.

Tempo a disposizione: 35 minuti.

2. RMI (Java)

Implementare un servizio remoto che esponga il metodo `cubo ()` definito nell'esercizio n.1.

Tempo a disposizione: 35 minuti.

3. Thread (C)

Una variabile intera x , inizializzata a 0, è condivisa tra due thread A e B. Entrambi i cercano di incrementare x , usando la variabile locale `cnt` per tener traccia del numero di volte in cui l'incremento ha avuto successo.

Quando x supera il valore 300, i thread terminano la loro esecuzione mostrando il proprio valore di `cnt`.

Il programma termina quando tutti i thread hanno terminato la loro esecuzione.

Implementare il sistema descritto proteggendo opportunamente la variabile x dagli accessi concorrenti ed introducendo dei tempi di attesa casuali tra le operazioni eseguite dai singoli thread.

Tempo a disposizione: 35 minuti.

4. Web Service e Servlet (Java)

Implementare un WS che esponga il metodo `cubo()` descritto nell'esercizio n.1.
Scrivere una servlet da usare per interrogare il servizio e mostrare il risultato ricevuto.

Tempo a disposizione: 40 minuti.