

## Esercitazione 22/11/2018

**N.B.: Consegnare tutti i file sorgente prodotti, denominati secondo lo schema: `server_A.c` o `Server_A.java`, `server_B.c` o `Server_B.java`...**

- A. Realizzare un server (in C o in Java) che si metta in ascolto sul **port 3333**, sul quale accetta richieste di connessione da un client.  
Sulla connessione stabilita, il server riceve una stringa, terminata dal carattere `\n` (si supponga che basti una sola operazione di lettura in ricezione da parte del server, per ricevere ciascuna stringa inviata dal client).  
Il server quindi scrive la stringa ricevuta sullo standard output e si rimette in attesa di un'altra eventuale stringa, o, se il cliente chiude la connessione, si rimette in attesa di richieste di connessione.  
Testare il server `server_A` usando telnet.
- B. Conservando la versione (A) del server, realizzarne un'ulteriore versione tale che ciascuna stringa ricevuta dal server, oltre ad essere scritta dal server sulla propria standard output, sia inviata nuovamente al client come risposta (senza alcuna modifica).  
Testare il server `server_B` usando telnet.
- C. Conservando la versione (B) del server, realizzarne un'ulteriore versione tale che il server implementi un metodo/funzione
- ```
int conta_vocali(s)
```
- che **per il momento, ai fini di questo quesito**, restituisce 1 per qualunque stringa argomento `s`.  
Il server invoca `conta_vocali(s)` su ciascuna stringa `s` ricevuta dal cliente e invia in risposta al client il valore intero restituito dalla funzione, **convertito nella stringa che ne è la rappresentazione decimale**. A questo scopo, in C si può usare la funzione `itoa()` o `printf()`, mentre in Java si può ricorrere a `println()` della classe `PrintWriter` o a `toString()`.  
Testare il server `server_C` usando telnet.
- D. Modificare `conta_vocali()` in modo che conti le lettere vocali presenti nella stringa argomento `s` e restituisca il valore conteggiato.
- E. (Opzionale) Realizzare un semplice client per testare i server creati ai punti precedenti.

**Ciascun quesito va affrontato solo dopo aver risolto il precedente, producendo del codice funzionante.**

**Organizzare i file sorgente prodotti, denominandoli secondo lo schema: `server_A.c` o `Server_A.java`, `server_B.c` o `Server_B.java`.**

**Impacchettare i file sorgenti in un file compresso denominato `nome_cognome_matricola.zip` e caricarli mediante il seguente link entro 60 minuti.**

**<https://goo.gl/cZAJpB>**

**Al termine dei 60 minuti non sarà più possibile caricare ulteriori files mediante il link di cui sopra.**