

**Esame di Sistemi Multimediali, Corso di Laurea in Informatica, Appello del 3 Dicembre 2001**

punteggio massimo 30/30

**Esercizio 1 (~4/30)**

Descrivere con un diagramma il protocollo “three way handshake”, evidenziando in ciascun segmento scambiato, i campi essenziali per il funzionamento del protocollo. Può essere aggiunto un commento (max 10 righe).

**Esercizio 2 (~4/30)**

Descrivere le differenze tra un router ed un repeater (max 10 righe di spiegazione).

**Esercizio 3 (~4/30)**

Indicare (con una o più croci) le caratteristiche che sono tipiche di una LAN.

Unico mezzo trasmissivo	
Comunicazione mediante circuito dedicato	
Comunicazione mediante trasmissione broadcast	
Tecnologia di livello 2 ed 1	
Tecnologia di livello Trasporto	
Comunicazione Host-to-Host	
Comunicazione Process-to-Process	

**Esercizio 4 (~4/30)**

Che cos'è il modo di funzionamento “promiscuo” di una scheda di rete ? (max 10 righe di spiegazione)

**Esercizio 5 (~2/30)**

Quanti possibili indirizzi possiede una rete di classe C ?

**Esercizio 6 (~2/30)**

Indicare (con una o più croci) qual è (o quali sono) gli identificatori di una connessione TCP.

(Indirizzo IP)	
(Indirizzo IP, Indirizzo IP)	
( portaTCP , portaTCP)	
( Indirizzo IP, portaTCP , Indirizzo IP)	
( portaTCP , Indirizzo IP, portaTCP)	
( Indirizzo IP, portaTCP , Indirizzo IP , portaTCP )	

**Esercizio 7 (~4/30)**

Quali funzioni svolgono (nel loro insieme) le funzioni **htons()**, **htonl()**, **ntohs()**, **ntohl()** ?

### Esercizio 8 (~6/30)

Indicare quali sono gli errori presenti nel progetto e nella implementazione della seguente applicazione. Deve essere considerato corretto che in caso di errore non risolvibile il server venga chiuso.

-----  
Un utente di un sistema linux (che non lavora come root ma come un comune utente) vuole implementare un'applicazione client server, basata su una connessione TCP, in cui il server svolge il compito di restituire un intero (di 4 byte) al client e poi chiude la connessione.

Per fare questo, decide di scrivere un server (server.c) che rimane in attesa sulla porta 23, ed un client che si collega a questo server e che riceve l'intero dalla connessione. Alla fine della comunicazione

Qui di seguito il codice del server.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>

int main(void)
{
    struct sockaddr_in Local, Cli;
    int sockfd, newsockfd, OptVal, len, ris;
    long int n=0;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) exit(1);

    OptVal = 1;
    ris = setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, (char *)&OptVal, sizeof(OptVal));
    if (ris < 0) exit(1);

    memset ( &Local, 0, sizeof(Local) );
    Local.sin_family = AF_INET;
    Local.sin_addr.s_addr = INADDR_ANY;
    Local.sin_port = 23;
    ris = bind(sockfd, (struct sockaddr*) &Local, sizeof(Local));
    if (ris < 0) exit(1);

    ris = listen(sockfd, 10 );
    if (ris == SOCKET_ERROR) exit(1);

    for(;;) {
        do {
            memset ( &Cli, 0, sizeof(Cli) );
            newsockfd = accept(sockfd, (struct sockaddr*) &Cli, &len);
        }
        while ( (newsockfd<0) && (errno==EINTR) );

        if(errno!=EINTR) exit(1);

        n++; // NUMERO INTERO DA TRASMETTERE
        write(newsockfd, &n, sizeof( long int) );
        close(newsockfd);
    }
    close(sockfd);
    return(0);
}
```