Nome Cognome Matricola_____

Reti di calcolatori

Prova scritta – 21 novembre 2012

Parte 1

Esercizio 1.1 [12 punti]
Si vuole realizzare un servizio Web per la raccolta dei dati sull'impegno di medici in operazioni chirurgiche. Il servizio prevede le seguenti pagine:

1. Un form in cui l'utente indica i medici dell'equipe che hanno effettuato l'intervento, il tipo di operazione e la sua durata in ore e frazioni di ora (es. *3.25* per *1 ora e 15 minuti*). Il tipo di operazione deve essere selezionato da un menu a scelta singola mentre i medici devono essere selezionabili da un menu a scelta multipla. Le opzioni dei menù devono essere generate dinamicamente a partire da array. Si assuma un array che contiene la lista ("Appendice", "Ernia", "Trapianto", "Menisco", "Frattura") per i tipi di operazioni e ("Kildare", "House", "Ippocrate", "Frankenstein", "Barnard") per i medici.
2. Una pagina di raccolta dei dati che memorizza sul server gli inserimenti fatti col form di cui al punto 1 nella stessa sessione di lavoro. L'impegno orario per un medico per lo stesso tipo di operazione si somma a quelli inseriti in precedenza.
3. Una pagina di riepilogo che stampa una tabella con l'impegno orario totale per ogni coppia medico-tipo di operazione. Per ogni medico si riportano poi tutti i tipi di operazione su cui ha l'impegno orario massimo.

Si scrivano il form e le due pagine di raccolta dati e riepilogo usando HTML/PHP. Si supponga di utilizzare il metodo POST nel form.

Esercizio 1.2 [4 punti]
Dato il seguente codice PHP

```
$a = "1*3";
$b = "$a.3";
$v = array('a'=>"$a+1",'b'=>'1$b');
$sum = $v['a']+$v['b']+$v['c'];
```

indicare quali sono le variabili definite nell'interprete dopo l'esecuzione del codice, con il loro tipo e valore.

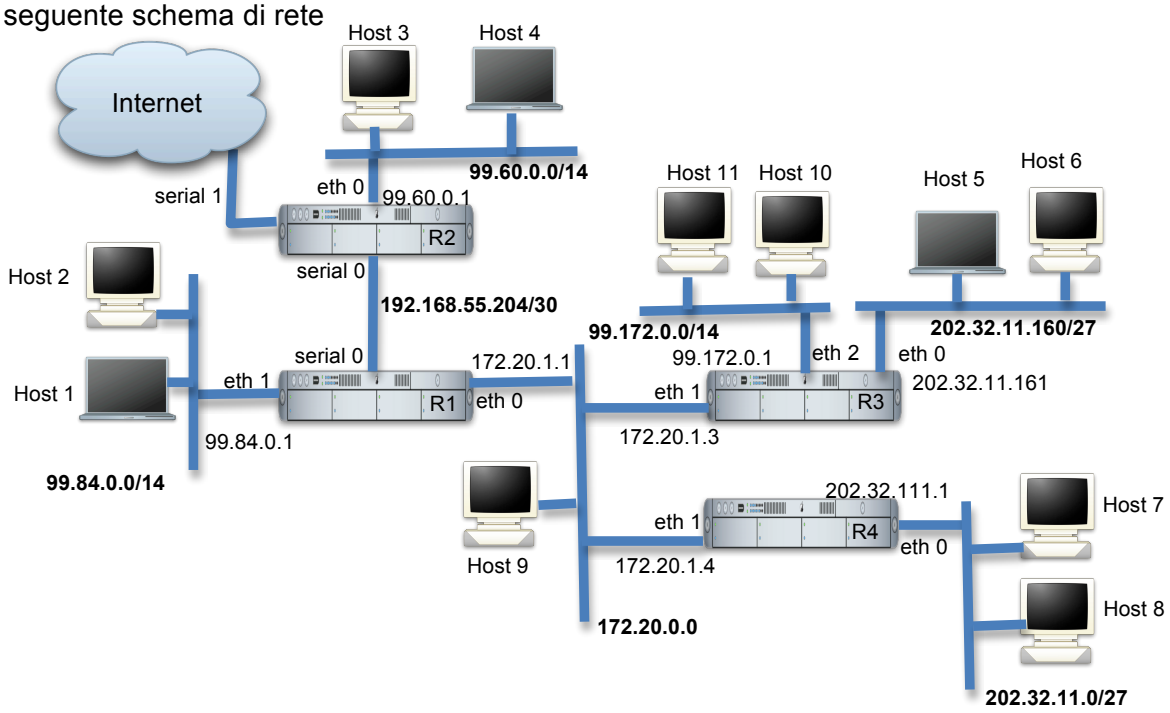
Domanda 1 [7 punti]
Spiegare cosa significa che il protocollo HTTP è senza stato (stateless), indicare le limitazioni che questo comporta e specificarte il supporto che è stato introdotto per gestire al livello di applicazione lo stato di transazioni HTTP fra due agenti.

Domanda 2 [7 punti] (solo per chi deve recuperare la parte 1)
Illustrare le principali funzionalità che riguardano i livelli di rete e di trasporto dell'archiettetura ISO-OSI.

Nome Cognome Matricola_____

Parte 2

Esercizio 2.1 [punti 6]
Dato il seguente schema di rete



scegliere la configurazione di rete dell'host 1 (IP, netmask e configurazione di routing) e indicare il contenuto delle tabelle di routing del router R3.

Esercizio 2.2 [punti 8]
Si vuole definire un formato per l'interscambio dati relativo la parco veicoli di una compagnia di noleggi. *Il file scambiato specifica la lista dei veicoli. Per ciasun veicolo si indica il modello (string), la targa (string), la data di acquisto (date) e la lista dei noleggi. Per ogni noleggio si specificano la data di inizio (date), il numero di giorni (integer) e il nominativo del cliente (string).* Si proponga la struttura XML necessaria, mostrando un esempio, e si scriva il file XML schema associato.

Esercizio 2.3 [punti 8]
Si scriva il codice javascript necessario ad implementare la seguente funzionalità in una pagina HTML: *cambiando la selezione su un menu a tendina (elemento <SELECT>) predefinito che riporta una serie di valori in punti per la dimensione del carattere (es. 8, 12, 16, 20), si cambia in modo corrispondente la dimensione dei caratteri dei paragrafi (elemento <P>) di una certa classe (attributo class). Suggerimenti: il cambiamento della selezione di un menu genera un evento di tipo "change" e la libreria DOM prevede il metodo getElementsByClassName.*

Esercizio 2.4 [punti 8]
Supponendo che un socket client TCP sia già stato opportunamente inizializzato (il suo identificatore è disponibile nella variabile `int sd`) e sia già avvenuta la connessione con il server, scrivere il codice per implementare il seguente protocollo di comunicazione.

1. Il client attende una stringa dal server.
2. Il client chiama la funzione predefinita `char *execute(char *instr)` passando come parametro la stringa ricevuta e invia al server la stringa ottenuta dalla funzione.
3. Il client attende una risposta dal server e, fino a che la risposta è la stringa "REPEAT", ripete la sequenza dal passo 2.
4. Quando ha ricevuto una stringa che non è "REPEAT", chiama la funzione `void store(char *reply)` passando come parametro la stringa ricevuta dal server e ripete la sequenza dal passo 1.