**Esercizio 1**

Si vuole realizzare un servizio Web per la gestione della prenotazione di un insieme di eventi da parte di un utente. Il servizio prevede le seguenti pagine:

1. un form in cui l’utente può inserire il nome dell’evento (una stringa) e il numero di persone per cui effettua la prenotazione. Il form permette l’inserimento di una prenotazione per volta e l’inserimento di più prenotazioni avviene inviando più volte il form in una stessa sessione.
2. una pagina di raccolta delle prenotazioni che memorizza sul server le prenotazioni fatte col form del punto 1. Si assume che se l’utente invia più di una volta il numero di prenotazioni per lo stesso evento nella stessa sessione, il numero di persone viene cumulato.
3. una pagina di riepilogo che stampa la lista delle prenotazioni, il totale del numero di persone e l’evento per cui è previsto il massimo numero di prenotazioni. Nel caso ci siano più eventi a cui corrisponde il valore massimo di prenotazioni si stampa il primo nella lista.

Si scrivano il form e le due pagine di prenotazione e riepilogo usando HTML/PHP. Si supponga di utilizzare il metodo GET nel form.

**Esercizio 2**

Si vuole realizzare un servizio Web per la gestione dell’acquisto di un insieme di prodotti da parte di un utente. Il servizio prevede le seguenti pagine:

1. un form in cui l’utente può inserire il nome del prodotto (una stringa) e il numero di pezzi che vuole acquistare. Il form permette l’inserimento di un acquisto per volta e l’inserimento di più acquisti avviene inviando più volte il form in una stessa sessione.
2. una pagina di raccolta delle richieste di acquisto che memorizza sul server gli acquisti fatti col form del punto 1. Si assume che se l’utente invia più di una volta il numero di pezzi per lo stesso prodotto nella stessa sessione, il numero di pezzi è uguale all’ultimo valore inserito.
3. una pagina di riepilogo che stampa la lista dei prodotti, il totale dei pezzi e il prodotto per cui è richiesto il massimo numero di pezzi. Nel caso ci siano più prodotti a cui corrisponde il valore massimo di pezzi si stampa il primo nella lista.

Si scrivano il form e le due pagine di prenotazione e riepilogo usando HTML/PHP. Si supponga di utilizzare il metodo GET nel form.

**Esercizio 3**

Si vuole realizzare un servizio Web per la gestione della raccolta di preferenze da parte di un utente su film. Il servizio prevede le seguenti pagine:

1. un form in cui l’utente può inserire il titolo del film (una stringa) e la sua valutazione numerica (si può assumere un valore nell’intervallo 1-5 ma non è necessario prevedere il codice per il controllo). Il form permette l’inserimento di una valutazione per volta e l’inserimento di più valutazioni avviene inviando più volte il form in una stessa sessione.
2. una pagina di raccolta delle valutazioni dei film che memorizza sul server le valutazioni fatte col form del punto 1. Si assume che se l’utente invia più di una volta una valutazione per lo stesso film nella stessa sessione, la valutazione è uguale al massimo dei valori inseriti.
3. una pagina di riepilogo che stampa la lista delle valutazioni, la media delle valutazioni e il film che ha ricevuto la valutazione massima. Nel caso ci siano più film a cui corrisponde il valore massimo di valutazione si stampa il primo nella lista.

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

**Esercizio 4**

Si vuole realizzare un servizio Web per la gestione della raccolta dei punteggi realizzati in videogiochi da parte di un utente. Il servizio prevede le seguenti pagine:

1. un form in cui l’utente può inserire il nome del videogioco (una stringa) e il suo punteggio numerica. Il form permette l’inserimento di un punteggio per volta e l’inserimento di più punteggi avviene inviando più volte il form in una stessa sessione.
2. una pagina di raccolta dei punteggi dei videogiochi che memorizza sul server i punteggi inseriti col form del punto 1. Si assume che se l’utente invia più di una volta un punteggio per lo stesso viodeogioco nella stessa sessione, il punteggio memorizzato è uguale al massimo dei valori inseriti.
3. una pagina di riepilogo che stampa la lista dei videogiochi con i punteggi, il massimo punteggio e il videogioco che ha il punteggio massimo. Nel caso ci siano più videogiochi a cui corrisponde il punteggio massimo si stampa il primo nella lista.

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

**Esercizio 5**

Si vuole realizzare un servizio Web per la gestione della raccolta dei risultati delle partite di calcio. Il servizio prevede le seguenti pagine:

1. un form in cui l'utente può inserire il risultato della partita selezionando la squadra di casa e quella ospite da un combobox generato dinamicamente a partire dalla lista delle squadre e specificando i gol delle due squadre in due campi di testo. Il form permette l'inserimento di un risultato per volta e l'inserimento di più risultati avviene inviando più volte il form in una stessa sessione.
2. una pagina di raccolta dei risultati che memorizza sul server le partite inserite col form del punto 1. Si assume che se l'utente invia più di una volta un risultato per la stessa partita nella stessa sessione, il punteggio memorizzato è l'ultimo inserito.
3. una pagina di riepilogo che stampa la classifica ordinata e calcolata assegnando 3 punti per la vittoria, 1 per il pareggio e 0 per la sconfitta.

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

**Esercizio 6**

Si vuole realizzare un servizio Web per la gestione dell'indicizzazione e la ricerca su un insieme di documenti di testo. Il servizio prevede le seguenti pagine:

1. un form in cui l'utente può inserire un documento specificandone il titol e il testo in un textbox. Il form permette l'inserimento di un documento per volta e l'inserimento di più documenti avviene inviando più volte il form in una stessa sessione.
2. una pagina di indicizzazione dei documenti che memorizza sul server i documenti inseriti col form del punto 1. Lo script deve creare l'indice che associa a ciascuna parola contenuta nei documenti, la lista dei documenti in cui compare (indice inverso) specificando per ogni documento il numero di volte che la parola vi compare (*Term Frequency* -TF). Inoltre per ogni parola deve essere memorizzato il numero di documenti in cui compare (*Document Frequency* - DF). Per ogni documento deve essere memorizzato il titolo e il testo contenuto. Per identificare le parole all'interno del testo si possono considerare come separatori la punteggiatura, gli spazi e altri caratteri non alfanumerici (parentesi, apici, ecc.). Le parole vengono indicizzate senza considerare differenze fra maiuscole e minuscole.
3. un form di ricerca che permette di inserire una sequenza di parole chiave separate dal carattere spazio.
4. una pagina che trova i risultati per l'interrogazione specificata nel form al punto 3. La lista dei risultati viene calcolata considerando tutti i documenti in cui compare almeno un termine di ricerca. Per tali documenti viene calcolato un punteggio che si ottiene sommando il peso TF-IDF di ogni parola dell'interrogazione trovata nel documento. Il peso TF-IDF è calcolato come TF/NT\*log(NDOC/DF) dove TF è il numero di volte che il termine compare nel documento, NT è il numero totale di termini nel documento, NDOC è il numero totale di documenti e DF è il numero di documenti che contiene il termine. La lista dei documenti che hanno un punteggio non nullo viene mostrata in ordine decrescente mostrando per ciascunoil titolo a cui è associato un link alla pagina di cui al punto 5.
5. una pagina che mostra il titolo e il testo del documento memorizzato nella sessione e il cui id è passato come parametro GET.

Si scrivano i form gli script richiesti ai punti 1-5 usando HTML/PHP.

**Esercizio 7**

Si vuole realizzare un servizio Web per la raccolta dei dati relativi alle temperature massime e minime di un insieme di stazioni metereologiche. Il servizio prevede le seguenti pagine:

1. Un form in cui l’utente può scegliere il nome della stazione da un menù a tendina e inserire la temperatura massima e minima. Il form permette l’inserimento di una misura per volta e le opzioni del menù a tendina devono essere generate dinamicamente a partire da un array con i nomi delle stazioni (si considerino “Trento”, ”Milano”, ”Torino”, ”Firenze”, “Bologna”, “Roma”, “Napoli”, “Bari”, “Messina”).
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. Ad ogni dato inserito deve essere associato anche il timestamp (si usi la funzione *time()* che fornisce il timestamp come numero di secondi fra il tempo presente e 1/1/1970 00:00:00 GMT).
3. Una pagina di riepilogo che stampa per ogni stazione la temperatura massima più alta e la media della temperatura minima del giorno corrente e dell’ultima settimana (si assuma che il giorno e la settimana corrente partano dal timestamp attuale meno 24\*60\*60 e 7\*24\*60\*60 rispettivamente). Infine supponendo che ciascuna stazione sia associata ad una zona (“Trento”=>”Nord”, ”Milano” => ”Nord”, ”Torino” => ”Nord”, ”Firenze” => ”Centro”, “Bologna” => ”Centro”, “Roma” => ”Centro”, “Napoli”=> ”Sud”, “Bari” => ”Sud”, “Messina” => ”Sud”), la pagina deve stampare la media, su tutti i dati disponibili, delle temperature massime e minime misurate per ciascuna zona.

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

**Esercizio 8**

Si vuole realizzare un servizio Web per la raccolta dei dati relativi alla pressione sanguigna massima e minima di un insieme di pazienti di controllo. Il servizio prevede le seguenti pagine:

1. Un form in cui l’utente può scegliere il nome del paziente da un menù a tendina e inserire la pressione massima e minima. Il form permette l’inserimento di una misura per volta e le opzioni del menù a tendina devono essere generate dinamicamente a partire da un array con i nomi dei pazienti (si considerino “Mario”, ”Giuseppe”, ”Filippo”, ”Maria”, “Rosa”, “Emma”, “Aldo”, “Leonardo”, “Marina”).
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. Ad ogni dato inserito deve essere associato anche il timestamp (si usi la funzione *time()* che fornisce il timestamp come numero di secondi fra il tempo presente e 1/1/1970 00:00:00 GMT).
3. Una pagina di riepilogo che stampa per ogni paziente la pressione massima più alta e la differenza media fra pressione massima e minima delle ultime 12 ore e dell’ultima giornata (si assuma che si parta dal timestamp attuale meno 12\*60\*60 e 24\*60\*60 rispettivamente). Infine supponendo che ciascun paziente sia associato ad una categoria (“Mario”=>”M-60”, ”Giuseppe” =>”M-60”, ”Filippo” =>”M-70”, ”Maria”=>”F-60”, “Rosa”=>”F-60”, “Emma” =>”F-70”, “Aldo” =>”M-70”, “Leonardo” =>”M-60”, “Marina” =>”M-70”) la pagina deve stampare la media, su tutti i dati disponibili, delle pressioni massime e minime misurate per ciascuna categoria.

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 9**

Si vuole realizzare un servizio Web per la raccolta delle disponibilità per l’organizzazione di una riunione. Il servizio prevede le seguenti pagine:

1. Un form in cui l’utente può scegliere i giorni della settimana e le fasce d’orario da due menù a scelta multipla e inserire il nome della persona in un campo testuale. Il form permette di specificare un insieme di disponibilità per un dato partecipante ad ogni invio. Le opzioni dei menù a scelta multipla devono essere generate dinamicamente a partire da due array con i nomi dei giorni della settimana e delle fasce di orario (si considerino le fasce “9-11”, ”11-13”, ”14-16”, ”16-18”).
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. Nel caso in cui si inseriscano disponibilità per la stessa persona in più richieste distinte, le disponibilità vengono aggiunte.
3. Una pagina di riepilogo che stampa una tabella con il numero di persone disponibili  per ogni opzione giorno-fascia oraria. Si riportano poi tutte le opzioni giorno-fascia oraria per cui c’è il massimo numero di persone disponibili. Nel caso in cui il massimo numero di persone disponibili non coincida con il totale dei partecipanti (la lista dei partecipanti coincide con l’insieme dei nomi inseriti col form), per ogni opzione si stampano le persone che non hanno dato disponibilità.

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.