Array

Esercizio 1

Implementare un servizio che permetta il login di utenti ad un sistema e la possibilità di visualizzare articoli. Il servizio riceve in ingresso, tramite GET, l'username e la password dell'utente. Poi verifica che siano corretti. In caso affermativo, procede come descritto in seguito, in caso negativo non esegue alcuna operazione.
La lista degli utenti è mantenuta staticamente in un array, così come la lista degli articoli. In base al ruolo dell'utente, il servizio mostra una schermata diversa. I possibili ruoli dell'utente e le relative schermate da mostrare sono:

- **amministratore**: il servizio mostra tutti gli utenti del sistema e i relativi ruoli, senza però mostrare le password degli utenti.
- **lettore**: il servizio mostra tutti gli articoli presenti nel sistema, senza il nome degli autori
- **scrittore**: il servizio permette di visualizzare tutti gli articoli scritti dall'utente

Soluzione

Per prima cosa, creiamo un array associativo $utente1, che contiene un utente:

```
<?php

// utente pippo
$utente1 = array('username' => 'pippo', 'password' => 'pippo123',
'ruolo' => 'amministratore');

?>
```

L'array associativo definisce le coppie (chiave, valore). Ripetiamo la stessa operazione per altri due utenti, a cui associamo gli altri ruoli possibili:

```
<?php

// utente pippo
$utente1 = array('username' => 'pippo', 'password' => 'pippo123',
'ruolo' => 'amministratore');

// utente pluto
$utente2 = array('username' => 'pluto', 'password' => 'pluto123',
'ruolo' => 'lettore');

// utente paperino
$utente3 = array('username' => 'paperino', 'password' => 'paperino123',
'ruolo' => 'scrittore');
```

```
?>
```

Aggiungiamo anche un utente 4, scrittore, che ci servirà dopo.

```
// utente paperino
$utente4 = array('username' => 'topolino', 'password' => 'topolino123',
'ruolo' => 'scrittore');
```

Guardando il codice, possiamo raggruppare i tre array $utente1, $utente2, $utente3 in un unico array multidimensionale $utenti:

```
<?php

$utenti = array($utente1, $utente2, $utente3, $utente4);
?>
```

Più semplicemente, possiamo eliminare le variabili $utente1, $utente2 e $utente3 e sostituire il codice precedente con:

```
// array degli utenti
$utenti = array();
$utenti[0] = array('username' => 'pippo', 'password' => 'pippo123',
'ruolo' => 'amministratore');
$utenti[1] = array('username' => 'pluto', 'password' => 'pluto123',
'ruolo' => 'lettore');
$utenti[2] = array('username' => 'paperino', 'password' =>
'paperino123', 'ruolo' => 'scrittore');
$utenti[3] = array('username' => 'topolino', 'password' =>
'topolino123', 'ruolo' => 'scrittore');
```

Ora creiamo un array $articoli contenente tutti gli articoli scritti dagli utenti 'scrittori'. Anche in questo caso, l'array è multidimensionale: per ogni scrittore, manteniamo la lista di tutti gli articoli. Questa volta, però, l'array $articoli è un array associativo, mentre l'array $utenti era un array numerico. In pratica l'array $articoli è un array associativo che contiene un array numerico, mentre l'array $utenti è un array numerico che contiene array associativi.

```
// array degli articoli
$articoli = array();
$articoli['paperino'] = array();
$articoli['paperino'][0] = "Primo articolo di Paperino";
$articoli['paperino'][1] = "Secondo articolo di Paperino";

$articoli['topolino'] = array();
$articoli['topolino'][0] = "Primo articolo di Topolino";
$articoli['topolino'][1] = "Secondo articolo di Topolino";
$articoli['topolino'][2] = "Terzo articolo di Topolino";
```

A questo punto possiamo passare alla lettura da GET delle credenziali dell'utente. Leggiamo dalla variabile $_GET i parametri *username* e *password*, verificando prima che essi siano stati settati dall'utente. Per verificare che i parametri siano stati settati, utilizziamo la funzione isset().

```
if(isset($_GET['username']) && isset($_GET['password']))
{
      // controlla login
      $username = $_GET['username'];
      $password = $_GET['password'];
} else
      echo "parametri mancanti";
```

Abbiamo utilizzato l'operatore logico && per verificare che entrambi i parametri siano stati settati. Se almeno uno dei due parametri non è stato settato, è eseguito il blocco else, per cui il programma stampa la scritta "parametri mancanti".
Procediamo ora con la scrittura del codice interno al blocco IF. Come prima cosa, occorre controllare che username e password siano corretti. Pertanto occorre scorrere l'array utenti e verificare che siano presenti username e password inseriti. Utilizziamo l'istruzione FOR:

```
for($i = 0; $i < count($utenti); $i++)
{
      if( ($utenti[$i]['username'] == $username) &&
            ($utenti[$i]['password'] == $password))
      {
            // abbiamo trovato l'utente!
      }
}
```

Se troviamo l'utente, allora dobbiamo eseguire le seguenti operazioni: 1) recuperare il ruolo dell'utente, 2) eseguire qualche operazione in base al ruolo, 3) uscire dal ciclo FOR.
Per quanto riguarda il recupero del ruolo, questo è memorizzato in $utenti[$i]['ruolo'], quindi basta accedere a questo elemento dell'array. Per quanto riguarda invece l'operazione da eseguire in base al ruolo, possiamo usare uno switch per discriminare i vari casi. Pertanto possiamo scrivere (dentro l'IF, che si trova dentro il FOR):

```
$ruolo = $utenti[$i]['ruolo'];
switch($ruolo)
{
      case 'amministratore':
            // operazioni amministratore
            break;
      case 'lettore':
            // operazioni lettore
            break;
      case 'scrittore':
            // operazioni scrittore
```

```
            break;
}
```

A questo punto occorre implementare il punto 3), cioè uscire dal ciclo FOR. Questo può essere fatto inserendo un break, dopo lo SWITCH:

```
for($i = 0; $i < count($utenti); $i++)
{
      if(   ($utenti[$i]['username'] == $username) &&
            ($utenti[$i]['password'] == $password))
      {
            // abbiamo trovato l'utente!
            $ruolo = $utenti[$i]['ruolo'];
            switch($ruolo)
            {
                  case 'amministratore':
                        // operazioni amministratore
                        break;
                  case 'lettore':
                        // operazioni lettore
                        break;
                  case 'scrittore':
                        // operazioni scrittore
                        break;
            }
            break;
      }
}
```

Ora passiamo all'implementazione delle operazioni in base ai ruoli. Partiamo dall'amministratore. Il servizio deve mostrare tutti gli utenti e i loro ruoli, senza però mostrare le password. Possiamo scorrere l'array $utenti tramite un ciclo FOR, così come abbiamo fatto prima, avendo l'accortezza di non utilizzare la variabile $i perché l'abbiamo già usata in un FOR esterno:

```
for($j = 0; $j < count($utenti); $j++)
{
      // istruzioni
}
```

All'interno del FOR, per scorrere gli elementi del record (che è un array associativo), possiamo usare l'istruzione FOREACH per scorrere l'array. L'istruzione FOREACH ha la seguente sintassi:

```
foreach($mioarray as $key => $value)
{
      // istruzioni
}
```

L'istruzione scorre l'array associativo $mioarray e, ad ogni iterazione, memorizza in $key la chiave del record e in $value il valore del record. Nel nostro caso, dobbiamo scorrere l'array $utenti[$j] e stampare la chiave e il valore:

```
for($j = 0; $j < count($utenti); $j++)
{
      echo "UTENTE $j <br>";
      foreach($utenti[$j] as $chiave => $valore)
      {
            echo "$chiave : $valore <br>";
      }
}
```

In questo modo il programma stampa anche la password. Noi vogliamo, invece, che la password non venga stampata. Per questo dobbiamo fare un controllo, prima di stampare la coppia chiave, valore. Se la chiave contiene la password, allora non va stampata.

```
for($j = 0; $j < count($utenti); $j++)
{
      echo "UTENTE $j <br>";
      foreach($utenti[$j] as $chiave => $valore)
      {
            if( $chiave == 'password')continue;
            echo "$chiave : $valore <br>";
      }
}
```

Il codice completo per l'utente amministratore è il seguente:

```
$ruolo = $utenti[$i]['ruolo'];
switch($ruolo)
{
      case 'amministratore':
            for($j = 0; $j < count($utenti); $j++)
            {
                  echo "UTENTE $j <br>";
                  foreach($utenti[$j] as $chiave => $valore)
                  {
                        if( $chiave == 'password')continue;
                        echo "$chiave : $valore <br>";
                  }
            }
            break;
      case 'lettore':
            // operazioni lettore
            break;
      case 'scrittore':
            // operazioni scrittore
```

```
            break;
}
```

Passiamo ora all'utente di tipo lettore. Il sistema deve mostrare tutti gli articoli presenti, senza mostrare il nome dell'autore. Dobbiamo scorrere l'array $articoli. Possiamo usare il ciclo FOREACH per scorrere l'array $articoli:

```
foreach($articoli as $chiave => $valore)
{
      // istruzioni
}
```

Nel nostro caso la chiave corrisponde al nome dell'autore, mentre il valore all'array degli articoli. Pertanto possiamo usare dei nomi di variabili più appropriati, in modo da rendere più leggibile il testo:

```
foreach($articoli as $autore => $array_articoli)
{
      // istruzioni
}
```

Per scorrere l'array degli articoli e stampare ogni singolo articolo, possiamo usare un ciclo FOR:

```
foreach($articoli as $autore => $array_articoli)
{
      for($j = 0; $j < count($array_articoli); $j++)
            echo $array_articoli[$j].”<br>”;
}
```

Per sapere la lunghezza di $array_articoli, abbiamo usato la funzione count(), che restituisce la lunghezza dell'articolo.
Il codice completo scritto fin qui è il seguente:

```
$ruolo = $utenti[$i][‘ruolo’];
switch($ruolo)
{
      case ‘amministratore’:
            for($j = 0; $j < count($utenti); $j++)
            {
                  echo “UTENTE $j <br>”;
                  foreach($utenti[$j] as $chiave => $valore)
                  {
                        if( $chiave == ‘password’)continue;
                        echo “$chiave : $valore <br>”;
                  }
            }
            break;
      case ‘lettore’:
```

```
            foreach($articoli as $autore => $array_articoli)
            {
                  for($j = 0; $j < count($array_articoli); $j++)
                        echo $array_articoli[$j].”<br>”;
            }
            break;
      case ‘scrittore’:
            // operazioni scrittore
            break;
}
```

Passiamo all’ultimo caso: lo scrittore. In questo caso, il sistema mostra solo gli articoli scritti dallo scrittore che si è loggato. L’utente corrente si trova nella variabile $username. Pertanto, occorre accedere all’array $articoli direttamente mettendo l’username corrente. L’array $articoli[$username] contiene la lista di tutti gli articoli dell’utente. Attraverso un ciclo FOR possiamo scorrere l’array $articoli[$username] e stampare tutti gli articoli:

```
for($j = 0; $j < count($articoli[$username); $j++)
      echo $articoli[$username][$j].”<br>”;
```

Il presente codice funziona sempre, perché in precedenza abbiamo verificato che $username fosse un utente registrato. Se però non avessimo fatto il controllo sugli username, avremmo dovuto aggiungere anche un controllo sull’esistenza dell’array $articoli[‘username’]:

```
if(isset($articoli[$username]))
{
      for($j = 0; $j < count($articoli[$username]); $j++)
            echo $articoli[$username][$j].”<br>”;
}
```

Riportiamo ora il codice completo dell’intero esercizio:

```
<?php

// array degli utenti
$utenti = array();
$utenti[0] = array(‘username’ => ‘pippo’, ‘password’ => ‘pippo123’,
‘ruolo’ => ‘amministratore’);
$utenti[1] = array(‘username’ => ‘pluto’, ‘password’ => ‘pluto123’,
‘ruolo’ => ‘lettore’);
$utenti[2] = array(‘username’ => ‘paperino’, ‘password’ =>
‘paperino123’, ‘ruolo’ => ‘scrittore’);
$utenti[3] = array(‘username’ => ‘topolino’, ‘password’ =>
‘topolino123’, ‘ruolo’ => ‘scrittore’);

// array degli articoli
$articoli = array();
$articoli[‘paperino’] = array();
```

```
$articoli['paperino'][0] = "Primo articolo di Paperino";
$articoli['paperino'][1] = "Secondo articolo di Paperino";

$articoli['topolino'] = array();
$articoli['topolino'][0] = "Primo articolo di Topolino";
$articoli['topolino'][1] = "Secondo articolo di Topolino";
$articoli['topolino'][2] = "Terzo articolo di Topolino";

if(isset($_GET['username']) && isset($_GET['password']))
{
      // controlla login
      $username = $_GET['username'];
      $password = $_GET['password'];

      for($i = 0; $i < count($utenti); $i++)
      {
            if(   ($utenti[$i]['username'] == $username) &&
                  ($utenti[$i]['password'] == $password))
            {
                  // abbiamo trovato l'utente!
                  $ruolo = $utenti[$i]['ruolo'];
                  switch($ruolo)
                  {
                        case 'amministratore':
                              for($j = 0; $j < count($utenti); $j++)
                              {
                                    echo "UTENTE $j <br>";
                                    foreach($utenti[$j] as $chiave =>
                                    $valore)
                                    {
                                          if( $chiave =='password')continue;
                                                echo "$chiave : $valore
                                                <br>";
                                    }
                              }
                              break;
                        case 'lettore':
                              foreach($articoli as $autore =>
                              $array_articoli)
                              {
                                    for($j = 0;
                                    $j < count($array_articoli); $j++)
                                          echo $array_articoli[$j]."<br>";
                              }
                              break;
                              case 'scrittore':
                                    if(isset($articoli[$username]))
                                    {
```

```
                                    for($j = 0;
                                    $j < count($articoli[$username]);
                                    $j++)
                                          echo
                                          $articoli[$username][$j].
                                          "<br>";
                              }
                              break;
                        }
                  }
                  break;
            }
}

} else
      echo "parametri mancanti";

?>
```

Esercizio 2

Modificare l'esercizio precedente in modo che, se il login fallisce, è mostrato il messaggio "nome utente o password errati". Scrivere inoltre del codice che aggiunge 10 utenti all'array $utenti e li mostra a video. Il nome di ogni utente corrisponde ad un numero random tra 1 e 10, mentre la password è data dalla stringa nomeutente123 (dove nomeutente è il numero estratto). Il ruolo dell'utente è determinato dall'estrazione di un altro numero random tra 0 e 2 (0 → amministratore, 1 → lettore, 2→ scrittore).