Version 2.1

Aucune question n'est piégée, répondez simplement et efficacement.

Bon courage!

Question 1 : algorithmie

a) Ecrivez, en PHP, une fonction optimale qui inverse le contenu d'une chaine.

Exemple « abcde » donnera « edcba »

Contraintes:

- Ne pas utiliser de fonctions de PHP (pas de strrev, strlen etc), uniquement vos propres fonctions

```
function reverseString($string)
{
   $i = 0;
   $rev = NULL;

   while ( $string[$i] ) {
         $i++;
   }
   $i--;

   while ( $string[$i] ) {
         $rev .= $string[$i];
         $i--;
   }

   return $rev;
}
```

NB : compte tenu des contraintes et des limitations du nombre d'opérations à effectuer, c'est l'approche qui me semble la plus efficace (O(n)). En effet, il faudrait

rajouter une operation de copiage de la chaine pour se prémunir des riques liés au caractère de terminaison.

Question 2 : décryptage d'un code php et optimisations MySQL

Un webmaster créé un jeu en ligne.

Pour attirer le plus grand nombre de joueurs potentiels, le jeu possède un mode de jeu gratuit et un autre payant plus avancé. Pour garder une trace des meilleurs résultats des différents modes *le webmaster* a créé le tableau `resultats` qui est rempli à la fin de chaque jeu effectué (partie):

```
-- Erreur sur default 'id_jeu' et default 'joueur';
CREATE TABLE `resultats` (
  `id_jeu` int(11) NOT NULL default '0' AUTO_INCREMENT,
  `joueur` varchar(64) NOT NULL default '',
  `mode_jeu` enum('abonne','gratuit') default 'gratuit',
  `points` int(11) NOT NULL default '0',
  `date_jeu` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (`id_jeu`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Sur la page d'accueil du site *le webmaster* affiche les 10 meilleurs scores des 7 derniers jours en fonction du mode de jeu sélectionné.

Pour cela il a créé la fonction suivante :

```
<?php
// mysql_connect() et mysql_select_db() ont déja été faits

/**
 * Gets sorted results from database
 * @return Array()
 */
```

```php
function get_classement ()
{
    // register_globals ON - WARNING
    $mode         = $_REQUEST['game_mode'];
    $periode      = $_REQUEST['period'];
    $nb_resultats = $_REQUEST['nb_game'];

    $classement   = array();
    $stats        = array();

    // SELECT 'resultats' within for a particular period '$periode'
    $R = mysql_query("SELECT * FROM resultats WHERE date_jeu + INTERVAL
$periode DAY < NOW();");

    if ( mysql_num_rows($R) > 0 ) {
        while ( $r = mysql_fetch_assoc($R) ) {
            if ( $r['mode_jeu'] == $mode ) {
                $classement[$r['points']] = $r;
            }
        }
        $stats['nb_joueurs'] = count($classement);

        if ( $stats['nb_joueurs'] > 0 ) {
            ksort($classement);
            $classement = array_slice($classement, 0, $nb_resultats);
        }
    }

    return array($classement , $stats);
}
?>
```

a) Le développeur a oublié de documenter son code… Le chef de projet vous demande de documenter le code. Veuillez documenter le code ci-dessus comme si vous en étiez l'auteur.

b) Si tous fonctionne bien au début, il se trouve que son jeu rencontre un énorme succès. La table grossit et dépasse désormais les 1 000 000 enregistrements. On observe alors de sérieux ralentissements voir une indisponibilité de la page qui appelle la fonction précédente. Le code développé par le développeur "naif" doit etre optimisé.

Que doit-on changer/modifier/ajouter dans :

- la structure de la table ;
CREATE INDEX date_jeux_index ON resultats(date_jeux);

- le code de la fonction ;

```php
<?php
// mysql_connect() et mysql_select_db() ont déja été faits

/**
 * Gets sorted results from database
 * @return Array()
 */
function get_classement ()
{
    $classement   = array();
    $stats        = array();

    // register_globals ON - WARNING
    $mode         = $_REQUEST['game_mode'];
    $periode      = $_REQUEST['period'];
    $nb_resultats = $_REQUEST['nb_game'];

    // Cleanup ... - WARNING
    $mode         =  mysql_real_escape_string($mode);
    $periode      =  mysql_real_escape_string($periode);
    $nb_resultats = mysql_real_escape_string($nb_resutats);

    // SELECT 'resultats' from within for a particular period '$periode'
    $R = mysql_query("SELECT * FROM resultats WHERE date_jeu + INTERVAL
$periode DAY < NOW() LIMIT $nb_resultats;");

    if ( mysql_num_rows($R) > 0 ) {
        while ( $r = mysql_fetch_assoc($R) ) {
            if ( $r['mode_jeu'] == $mode ) {
                $classement[$r['points']] = $r;
            }
        }
        $stats['nb_joueurs'] = count($classement);
        ksort($classement);

        return array($classement , $stats);
```

```
    }
    return array();
}
?>
```

Pour que le site se remette à fonctionner DE MANIERE OPTIMALE.

Question 3 : Bases de données

ATTENTION considérer qu'aucune optimisation vue dans la question précédente n'a été appliquée (la structure de table est conservée)

a) On décide d'afficher maintenant la liste des points de l'utilisateur « John doe » selon un rapport quotidien. Ecrire LA requête qui permettrait d'afficher JOUR PAR JOUR l'évolution des points de cet utilisateur.

La requête doit retourner un résultat sous la forme :

date	**points**
2010-01-01	40
2010-01-02	22
2010-01-05	24
2010-01-08	1

ATTENTION au format du champ date_jeu
ATTENTION un joueur peut jouer plusieurs fois par jour

SELECT DATE_FORMAT(date_jeu, '%Y-%m-%d') AS 'date', sum('points') AS 'points' FROM resultats R WHERE R.joueur = 'john doe' GROUP BY R.date_jeu ORDER BY R.date_jeu ASC;

b) On décide maintenant d'ajouter une table « tricheur » qui enregistre les joueurs réputés comme tricheurs.

```
CREATE TABLE `tricheur` (
  `joueur` varchar(64) NOT NULL default '',
  `comment` TEXT NOT NULL,
  PRIMARY KEY  (`joueur`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Ecrire la requête qui permet de récupérer toutes les parties (jeux) pour lesquelles il y a eu tricherie

-- On poourrait stocker les parties grugées sous forme d'une CSV dans 'comments'
SELECT id_jeu FROM resultats WHERE id_jeu IN (SELECT comment FROM tricheur);

c) Ecrire la requête qui permet de récupérer toutes les parties (jeux) pour lesquelles il n'y a PAS eu tricherie.

SELECT id_jeu FROM resultats WHERE id_jeu NOT IN (SELECT comment FROM tricheur);

Question 4 : regexp

Soit la REGEXP

```
/<([A-Z][A-Z0-9]*)[^>]*>.*</\1>/i
```

a) Quel type de texte match cette *regular expression*? (Ne pas donner le détail de chaque morceau de la regexp, mais essayer de trouver quel motif est matché par la regexp)

Elle previent l'insertion de balises html ou de scripts

b) Ecrire un exemple de ligne qui sera trouvée par la regexp

<script123>docmuent.write('foo')</sCriPt>

c) Ecrivez une regexp SIMPLE qui permet de valider une date au format:

2010-10-01 12:42:42

([0-9]{4})-([0-9]{2})-([0-9]{2})

Question 5 : Objet

On implémente le code suivant :

```php
<?php
abstract class object1
{
    abstract public function foo1();

    private function _foo2()
    {
        echo 'foo2';
    }

    public function foo3()
    {
        echo 'foo3';
    }

    public function foo4()
    {
        return "foo4";
    }
}

final class object2 extends object1
{
    public function foo5()
    {
        parent::_foo2();
    }

    public function foo4()
    {
        return "Here is " . parent::foo4();
    }
}

$obj1 = new object1();

$obj1->foo3();

$obj2 = new object2();

$obj2->foo5();
echo $obj2->foo4();
```

a) Quelles sont les erreurs qui se sont glissées dans ce code (x3)

Instanciation d'une classe abastraite
Appel d'une méthode d'un objet non existant (la classe abstraite)
Problème d'encapsulation, appel à une méthode privée (on aurait du la déclarer en 'protected')

b) Qu'affiche le code si on corrige les différentes erreurs ?
'here is foo 4' ou avec correction l'encapsualtion 'foo2Here is foo4'

c) Soit le code suivant :

```php
<?php

class MyClass
{
    static protected $_instance = null;

    final private function __construct()
    {
        //do something
    }

    public static function getInstance()
    {
        if (self::$_instance === null)
        {
            self::$_instance = new self();
        }
```

```
        return self::$_instance;
    }
}
```

Décrivez la particularité que contient cette classe:
Il s'agit d'un DESIGN PATTERN de type SINGLETON. On l'utilise très fréquemment notamment pour limiter les instances de connexion à une bases de données l'ors de l'éxécution d'un script.

Question 6 : Culture générale PHP/MySQL

a) Quelle faille de sécurité est présente dans le code suivant ?
Quel en est le danger ?
Comment s'en protège-t-on ?
Corrigez le code pour avoir une version plus sécurisée

```php
<?php

// Il y a un problème d'injection SQL
// On peut s'en protéger avec la fonction mysql_real_escape_string
$id = mysql_real_escape_string($_POST['id']);
echo $db->query('SELECT * FROM data WHERE id = ' . $id);
```

b) Quelle faille de sécurité est présente dans le code suivant
Quel en est le danger ?
Comment s'en protège-t-on ?
Corrigez le code pour avoir une version plus sécurisée

```php
<?php
// Un attaquant pourrait demander à lister des fichiers, non autorisés,
// voir même sensibles
// par exemple en renseignant $_POST['user_id'] avec * etc..
$path = './tmp';
$command = escapeshellarg($path . '/' . $_POST['user_id'] . '.txt');
exec('cat ' . $path . '/' . $_POST['user_id'] . '.txt');
```

c) Quelle fonction permet de dupliquer le processus PHP courant. Expliquer brièvement son fonctionnement.

pcntl_fork()

Elle crée une copie exacte (code et données) d'un processus père. Le père et le fils ont alors une éxécution concurrente. Il en découle une possibilité d'arborescence des processus.

FIN