

Dès le début de l'épreuve, indiquez votre numéro de place sur le formulaire
Il vous faut impérativement utiliser ce formulaire pour répondre à certains
exercices. La place réservée aux réponses est limitée mais suffisante. Vos
réponses doivent donc être concises mais précises. Ce formulaire est à insérer
dans votre copie, comme un intercalaire.

Exercice 1 : (Répondre sur le formulaire) Soit la feuille HTML suivante :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta content="text/html; charset=UTF-8" http-equiv="content-type" /><title>Qstyle</title>

<style>
→ 011 -h1.special {color:red;}
→ 002 -div h1 {color:green;}
→ 012 -div > h1:first-child {color:grey;}
→ 001 h1 {color:blue;}
→ 002 xdiv > h1 {color:magenta;}
→ 100 -#repere1 {color:cyan;}
→ 101 xdiv#repere2 {color:orange;}
→ 002 -div + h1 {color:yellow;}
</style>
</head>
<body>
<pre> <br /> <h1> Bloc A </h1> <br /> </pre>
<h1 class="special"> Bloc B </h1>
<div>
<h1> Bloc C </h1>
<h1> Bloc D </h1>
<h1 class="special"> Bloc E </h1>
</div>
<div>
<h1 class="special" id="repere1"> Bloc F </h1>
<pre><h1> Bloc G </h1> </pre>
</div>
<h1> Bloc H </h1>
<h1> Bloc I </h1>
<h1 id="repere2"> Bloc J </h1>
<div> Bloc K </div>
</body>
</html>
```

Indiquez la couleur d'écriture de chacun des «blocs». (On suppose que la couleur d'écriture par défaut est le noir et qu'il n'y a pas de paramétrage de style particulier côté utilisateur).

Exercice 2 : (Répondre sur le formulaire) Écrire le code source XHTML et une feuille de style associée permettant d'obtenir le tableau affiché dans la figure ci-contre, avec cette présentation (il sera tenu compte de la clarté de l'écriture et de la présentation). NB : la couleur des cases des coins est jaune.

x	1	2	y
A	Centre		A'
B			B'
z	Bas		t

Exercice 3 : (Répondre sur la copie) Un document XHTML est supposé contenir plusieurs chapitres. Le contenu de chaque chapitre est encapsulé dans un élément `div` auquel on a attribué une classe nommée `chapitre`. Chacun de ces éléments `div` est supposé également doté d'un attribut `id`.

On souhaite qu'un navigateur exécutant du code javascript insère systématiquement à la fin de chaque chapitre un lien permettant de remonter au début du chapitre.

Question 1 : Écrire la fonction javascript `ajouteliens()` qui ajoute en fin de chaque chapitre le lien (un élément `a`) permettant de revenir au début. Le texte de chaque lien sera «Retour au début du chapitre».

Rappel : l'URL désignant le début d'un élément d'identifiant `x` est `"#x"`

Question 2 : Comment associer ce code javascript au fichier HTML et faire exécuter la fonction lors du chargement de la page (sans insertion de code javascript dans le HTML).

Exercice 4 : De nombreuses questions ou sous-questions sont indépendantes. Il vous est recommandé de toutes les lire dès le début.

On considère une base de données mémorisant des morceaux musicaux. Chaque morceau est supposé appartenir à un album. Il porte un numéro (son rang) au sein de son album. Dans chaque album, la numérotation commence à 1. On supposera que les albums portent des noms tous distincts. **Important : par contre, il peut y avoir homonymie pour les titres de morceaux.**

La base comporte pour l'instant 2 tables :

- **Morceaux** : y sont définis le **titre** (chaîne) , la **longueur** en secondes (entier), le nom de l' **album** (chaîne) et le **rang** du morceau au sein de l'album (entier ≥ 1).

- **Albums** : le **nom** (chaîne), le nom de l'**artiste** principal (chaîne), la date de **sortie** (date).

La base est supposée cohérente du point de vue de la numérotation des morceaux : il ne peut y avoir deux morceaux portant le même rang au sein du même album.

Question 1 : (Répondre sur le formulaire)

Voici une liste de tables qu'un utilisateur de la base veut afficher. Pour chacune d'elles, vous donnerez la requête SQL permettant de l'obtenir :

- 1.a Les titres et longueurs des morceaux de moins de 180 secondes (au sens large).
- 1.b Les titres des morceaux et le nom de leur album, sans préciser d'ordre.
- 1.c Dans quel ordre les lignes s'afficheront-elles ?
- 1.d Modifier la requête pour assurer que la table s'affiche par ordre alphabétique des albums puis, pour un même album, par rang croissant.
- 1.e Les titres et longueurs des morceaux de l'artiste 'UneTelle' et leur album.
- 1.f Le ou les artistes interprétant les morceaux de titre "xxx". Chaque artiste doit apparaître **au plus** une fois.
- 1.g Le ou les artistes interprétant les morceaux de titre "xxx", avec pour chaque interprétation, le nom de l'album et le rang au sein de cet album.
- 1.h Les noms des albums et leur longueur totale.
- 1.i Les noms des albums comportant strictement moins de 3 morceaux.

Question 2 : (Répondre sur la copie)

Il vous est demandé de réaliser une interface Web qui permet de saisir le nom d'un artiste puis qui affiche la discographie de cet artiste, à savoir la liste de ses albums (classés par ordre de date de parution). Cette interface est décomposée en 2 documents : l'un contient un formulaire de saisie et l'autre affiche le résultat de la demande.

(Important pour la suite : Il n'est pas utile, et même pas recommandé, de vous préoccuper ici de mise en page trop élaborée –bien-sûr en pratique ce serait nécessaire, mais pas dans le cadre de ce travail écrit–)

- 2.a Écrire le code de la page de saisie.
- 2.b Puis celui de la page permettant l'affichage des résultats. Vous distinguerez clairement la ou les requêtes SQL utilisées au sein du code. Vous traiterez les différents cas d'erreurs.