

Épreuve de 2^{ème} session

L'examen est sur 20 points et comprend 6 pages. Les quatre exercices sont indépendants. Les supports de cours, énoncés et notes manuscrites de TD sont les seuls documents autorisés. Les réponses doivent être soigneusement justifiées. L'utilisation de téléphones portables ou calculatrices est interdite. Le barème est donné uniquement à titre indicatif. Un aide mémoire HTML et PHP est donné en page 6.

Les instructions suivantes doivent être respectées scrupuleusement, sous peine de non-correction :

- Reporter sur **toutes** les copies le numéro d'anonymat imprimé sur la copie principale. Cette copie **doit** être rendue, même blanche.
- Chaque copie doit être numérotée et doit mentionner le nombre total de copies rendues (1/3, 2/3, ...). Si une seule copie est rendue, la numéroter 1/1. Les copies non numérotées **ne seront pas corrigées**.
- Toutes les réponses doivent être rédigées **au stylo**, sur des copies. Les parties rédigées au crayon de papier ou sur les feuilles de brouillons **ne seront pas corrigées**.
- Les copies doivent être anonymées, mais uniquement après vérification d'identité par un surveillant et signature de la liste d'émargement.

Questions de cours (3 points)

1. (1,5 point) À quoi sert un *système de fichiers* ?

Réponse: Un *système de fichiers* est une convention d'organisation des fichiers sur un support de stockage (disque dur, clé usb, ...). Différents systèmes de fichiers mettent l'accent sur différents points critiques :

- efficacité d'accès : on veut pouvoir accéder rapidement aux fichiers
- efficacité de stockage : on veut perdre le moins d'espace possible
- gestion de versions : on veut pouvoir revenir « en arrière » à une version précédente d'un fichier
- sécurité : on veut stocker les données de manière cryptée
- ...

2. (1,5 point) Qu'est-ce qu'un *cookie* ?

Réponse: Un *cookie* est un fichier stocké par le navigateur Web, sur la machine d'un client, à la demande d'un serveur web (site web). Le site peut utiliser ce fichier pour stocker des informations et les utiliser entre deux visites.

Programmation PHP, HTML et CSS (8,5 points)

Conseil : le problème rapporte beaucoup de points. Les questions sont toutes indépendantes. Pour répondre à une question vous avez le droit d'utiliser les fonctions définies dans les questions précédentes, même si vous n'y avez pas répondu. Si vous ne savez pas écrire le code PHP, écrivez du pseudo-code ou décrivez la procédure en Français.

On souhaite réaliser un petit formulaire Web demandant à un utilisateur de rentrer un nombre à 4 chiffres et produisant un affichage « digital » de ce nombre. Le programme Web comporte 2 pages, `saisie.html` écrite purement en HTML et `digital.php` écrite en HTML+PHP. Un exemple d'utilisation est donné à la Figure 1.

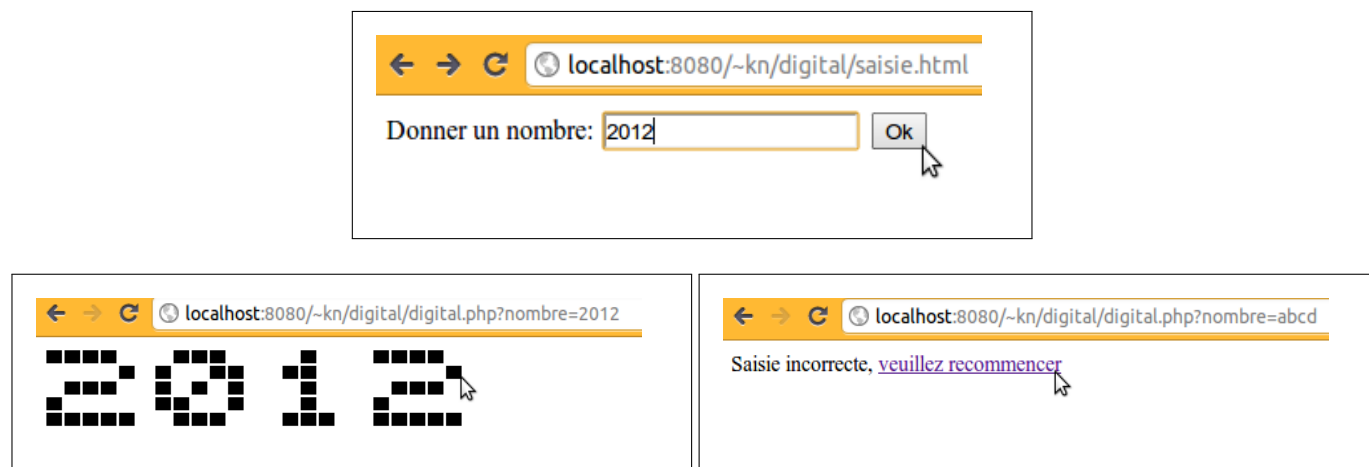


FIGURE 1 – Utilisation de la page `saisie.html`, rendu graphique et affichage d'erreur

Sur la page `saisie.html`, l'utilisateur rentre un nombre dans le formulaire et valide en pressant le bouton « Ok ». Ce nombre est passé à la page `digital.php` au moyen de la méthode `get`. Deux choses peuvent alors ce produire :

1. Le nombre est une chaîne de caractères, composée **exactement** de quatre chiffres (en particulier, avec des zéros initiaux si besoin). Dans ce cas, l'affichage digital a lieu. Par exemple, la chaîne « 123 » est invalide alors que la chaîne « 0009 » est acceptée.
2. La chaîne de caractères est d'un autre format, et un message d'erreur contenant un lien vers la page de saisie est affiché.

Étant donné une chaîne de caractères de 4 chiffres, l'affichage se déroule de la manière suivante :

- on dispose de dix fichiers (`0.txt`, `1.txt`, ..., `9.txt`) contenant une description du chiffre à l'aide d'espaces et de « # ». Chaque fichier contient 5 lignes de 5 caractères chacune. Le contenu des fichiers est donné à la Figure 2.
- on effectue le rendu en créant une table HTML (élément `table`, contenant des éléments `tr` eux-même contenant des éléments `td`). La case d'une table est toujours vide (pas de texte) mais elle possède un attribut `class` valant soit `on` soit `off`. On utilise ensuite une feuille de style CSS pour définir exactement les styles `on` et `off`

# # # #	# # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #
# # # #	# # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #
# # # #	# # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #
# # # #	# # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #
# # # #	# # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #	# # # #

FIGURE 2 – Contenu des 10 fichiers de description

Par exemple, pour l'affichage de la chaîne « 2012 » représenté à la Figure 1, `digital.php` produit le fichier HTML donné à la Figure 3. Dans ce code source, n'est représentée que la première ligne de « pixels » pour la chaîne « 2012 ». En effet, la séquence « `on`, `on`, `on`, `on`, `off` » correspond à la première ligne « `####` » du « 2 ». Le « `off` » suivant représente un blanc, suivit de la séquence « `off`, `on`, `on`, `on`, `off` » pour « `###` » représentant la première ligne du « 0 », etc.

Questions

1. (1,5 point) Donner le code HTML de la page `saisie.html`. Outre les balises `html`, `body`, ..., cette page contient un formulaire HTML ayant pour action la page `digital.php` et utilisant la méthode

```

<html>
<head>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<table>
<tr>
<td class="on" /><td class="on" /><td class="on" /><td class="on"/><td class="off" />
<td class="off" />

<td class="off" /><td class="on" /><td class="on" /><td class="on" /><td class="off" />
<td class="off" />

<td class="off" /><td class="off" /><td class="on" /><td class="off" /><td class="off" />
<td class="off" />

<td class="on" /><td class="on" /><td class="on" /><td class="on" /><td class="off" />
<td class="off" />
</tr>

...

</table>
</body>
</html>

```

FIGURE 3 – Code HTML produit par `digital.php` pour le nombre 2012

`get`. Ce formulaire contient le texte « Donner un nombre », un champs de saisie dont l'attribut `name` vaut `nombre` et un bouton de validation dont le label est `Ok`.

2. (1,5 point) On cherche maintenant à écrire le fichier `digital.php`. Donner un bout de code PHP qui teste la valeur du paramètre `nombre` passé par la méthode `get`. Si ce paramètre est une chaîne de caractères composée d'exactly quatre chiffres, alors la stocker dans une variable `$num` et appeler la fonction `draw($num)`; Sinon afficher le message d'erreur de la Figure 1 (avec le lien hypertexte renvoyant sur la page de saisie).
3. (4 points) Le but est maintenant d'écrire le code de la fonction `draw`. On procède par étapes :
 - (a) (1,5 points) Écrire une fonction `load_chars()` qui ne prend pas d'argument et renvoie un tableau `$tab` à 10 entrées (de 0 à 9) chacune contenant un tableau. Pour i entre 0 et 9, `$tab[i]` est lui même un tableau de 5 entrées (de 0 à 4), chaque entrée j étant une chaîne de caractères correspondant à la $j^{\text{ème}}$ ligne du fichier `i.txt`. En d'autres termes, `load_chars()` doit renvoyer un tableau de la forme :

```

( 0 => ( 0 => " ### ",
        1 => "#  ## ",
        2 => "# #  # ",
        3 => "##   # ",
        4 => " ###  "),
  1 => ( 0 => "  #  ",
        1 => " ##   ",
        ...
  )

```

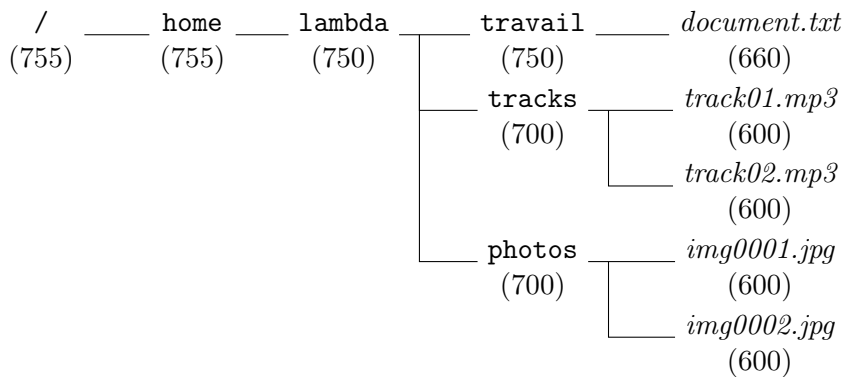
- (b) (1,5 points) écrire une fonction `draw_line($num, $l, $tab)` prenant comme argument une chaîne de caractères à 4 chiffres (`$num`), un numéro de ligne (`$l`, compris entre 0 et 4) et un tableau tel que renvoyé par `load_chars()` (`$tab`) et affiche un élément `<tr> </tr>` dont le contenu est une suite d'éléments `<td class="on" />` ou `<td class="off"/>`. Par exemple si `$num` vaut "2012" et `$l` vaut 0, alors on affichera un élément `<tr>...</tr>` identique à celui de la Figure 3.

- (c) (1 point) Donner finalement le code de la fonction `draw($num)` qui prend en argument une chaîne de caractères, charge un tableau à l'aide de la fonction `load_chars()` et affiche cette chaîne au moyen d'un élément `<table>` dont les lignes sont produites par la fonction `draw_line`.
4. (1,5 point) Donner le code CSS du fichier `style.css` permettant un affichage correct du tableau :
- les éléments `table` n'ont pas de bordure
 - les éléments `td` n'ont pas de bordure, et font dix pixels de long et dix pixels de large
 - les éléments `td` de classe `off` ont la couleur de fond à « blanc » (`white`)
 - les éléments `td` de classe `on` ont la couleur de fond à « noir » (`black`)
 - tous les autres attributs sont laissés à leur valeur par défaut.

Réponse: Voir les fichiers joints au corrigé.

Ligne de commande Unix (4,5 points)

On considère l'arborescence suivante (les noms en police **fixe** représentent des répertoires et les noms en *italique* représentent des fichiers). Les permissions sont données en octal, entre parenthèses :



Les répertoires `/` et `home` appartiennent à l'utilisateur `root` et au groupe `root`. Les autres fichiers et répertoires appartiennent à l'utilisateur `lambda` et au groupe `etud`. On suppose que l'utilisateur `lambda` est devant un *shell* et que le répertoire courant est `/home/lambda`.

Questions

1. (2,5 points) Pour chacun des fichiers ou répertoires : `lambda`, `travail`, `document.txt`, `photos` et `img0001.jpg` dire quels sont les droits pour l'utilisateur, le groupe et les autres (lecture, écriture ou exécution) (0,5 point/réponse).

Réponse:

Fichier	Utilisateur	Groupe	Autres
lambda	L, E, X	L, E	
travail	L, E, X	L, E	
document.txt	L, E	L, E	
photos	L, E, X		
img0001.txt	L, E		

L = Lecture, E = Écriture, X = exécution

2. (0,5 point) Donner une commande permettant de lister le contenu du répertoire `tracks`.

Réponse: 1s tracks

3. (0,5 point) Donner une commande permettant d'effacer le répertoire `tracks` et son contenu.

Réponse: `rm -rf tracks`

4. (0,5 point) Donner une commande permettant de lister le répertoire parent.

Réponse: `ls ..`

5. (0,5 point) Donner une commande permettant de donner au répertoire **photos** les mêmes permissions au groupe qu'à l'utilisateur (*i.e.* 770).

Réponse: `chmod 770 photos`

Réseaux, TCP/IP (4 points)

L'utilisateur **lambda** lance un navigateur Internet vers l'URL `http://www.u-psud.fr`. Le navigateur affiche une erreur de connexion. L'utilisateur **lambda** ouvre un terminal et décide de voir d'où vient le problème. Il lance la commande `ifconfig` et apprend que l'adresse IP de sa machine est 91.121.200.14. Il sait aussi qu'il est en adressage direct. Son DNS a pour adresse 213.186.33.99. La commande `ping 213.186.33.99` échoue, alors que la commande `ping 8.8.8.8` réussit.

Questions

1. (1 point) À quelle classe appartient l'adresse IP de la machine de l'utilisateur **lambda** ?

Réponse: La machine de l'utilisateur **lambda** a pour IP : 91.121.200.14. L'adresse est comprise entre 0.0.0.0 et 127.255.255.255, c'est donc une *adresse de classe A*.

2. (1 point) Que signifie l'échec de la commande `ping 213.186.33.99`.

Réponse: L'échec de la commande signifie que la *route* entre la machine de l'utilisateur **lambda** et la machine faisant office de DNS est interrompue (soit par ce que le lien est physiquement rompu, soit parce que la machine 213.186.33.99 est hors service).

3. (2 points) Quel est la cause de l'erreur affichée par le navigateur Internet ? (décrire **en détail** les étapes de connexion lorsqu'on accède à un site web, et dire à quel niveau la connexion échoue).

Réponse:

- (a) L'utilisateur rentre l'adresse `http://www.u-psud.fr` ;
- (b) Le navigateur internet contacte le serveur DNS pour effectuer une résolution d'adresse pour l'hôte `www.u-psud.fr` ;
- (c) Le DNS est inaccessible, la résolution de nom ne peut avoir lieu (l'erreur se situe ici) ;
- (d) Le navigateur ne peut se connecter à la machine `www.u-psud.fr` (car il n'arrive pas à déterminer son adresse) et affiche un message d'erreur à l'utilisateur.

Aide mémoire HTML et PHP

HTML

Élément	Description
<code> texte </code>	Affiche <i>texte</i> comme un lien hypertexte dont la cible est <i>url</i> .
<code> texte </code>	Affiche <i>texte</i> en gras.
<code><body>...</body></code>	Élément principal contenant le code HTML de la page.
<code>
</code>	Retour à la ligne.
<code><button type="submit"> texte </button></code>	Bouton dans un formulaire (à utiliser sous un form). <i>texte</i> est utilisé comme étiquette du bouton.
<code><form method="m" action="url">...</form></code>	Formulaire. <i>url</i> est l'adresse cible du formulaire. <i>m</i> doit valoir "post" ou "get".
<code><head>...</head></code>	En-tête du document HTML contenant le titre et d'autres méta-données
<code><html>...</html></code>	Racine du document, ouverte en premier et fermée en dernier.
<code><input type="text" name="n" /></code>	Zone de saisie dans un formulaire (à utiliser sous un form). Le nom du paramètre associé est <i>n</i> .
<code><option value="v"> texte </option></code>	Choix dans une liste déroulante (doit être utilisé sous un select). La valeur du choix est <i>v</i> et le texte affiché dans la liste est <i>texte</i> .
<code><select name="n"> ... </select></code>	Liste déroulante (contient une suite de option et doit apparaître sous un form). Le nom du paramètre est <i>n</i> .
<code><table>...</table></code>	Contient la définition d'une table
<code><tr>...</tr></code>	Contient la définition d'une ligne d'une table
<code><td>...</td></code>	Contient la définition d'une case d'une table

PHP

Fonction ou variable	Description
.	Concaténation de chaînes
array()	Création d'un nouveau tableau vide
count(\$a)	Renvoie le nombre d'éléments du tableau \$a
echo <i>e</i>	Écrit la valeur de l'expression <i>e</i> dans le document HTML
file(\$f)	Renvoie un tableau indicé par 0, ..., $n - 1$ contenant les n lignes du fichier dont le nom est passé en argument. Les retour à la lignes finaux sont conservés.
preg_match(\$r, \$s)	Renvoie le nombre d'occurences de l'expression régulière \$r dans la chaîne \$s.
\$_GET	Tableau global contenant les paramètres de formulaires passés avec la méthode get.