

## RÈGLEMENT SUR LE PLAGIAT

*Extrait du règlement disciplinaire sur le plagiat ou la fraude de l'université de Montréal*

Constitue un plagiat: faire exécuter son travail par un autre; utiliser, sans le mentionner, le travail d'autrui; **échanger des informations lors d'un examen**; falsifier des documents. Le plagiat est passible de sanctions allant jusqu'à l'exclusion du programme.

## CONSIGNES

- Une feuille de notes US-Letter recto-verso est permise.
  - Pas d'appareil électronique (vous pouvez cependant l'utiliser pour consulter l'heure)
  - Le barème est donné à titre indicatif sur 44 points. Votre note sera normalisée pour la ramener à un maximum de 25.
- 

(8) 1. Questions de cours

- (a) Qui a développé la première version du langage HTML?
- (b) Indiquez un élément HTML de contenu vide ayant au moins un attribut obligatoire que vous indiquerez.
- (c) À quoi sert l'attribut `title` en HTML ?
- (d) Quelle est la différence entre le web et l'internet?
- (e) Sur quelle technologie sont basés les cookies ?
- (f) Qui a développé le protocole HTTP?
- (g) Expliquez le rôle d'une requête GET et d'une requête POST et précisez en quoi elles diffèrent le plus.
- (h) Considérez la page XHTML de la question 2 et supposez qu'elle se trouve à l'URL `http://www.iro.umontreal.ca/~pift3225/index.html`. Combien de requêtes HTTP seront faites par votre navigateur si vous lui demandez l'URL `http://www.iro.umontreal.ca/~pift3225` ? Indiquez la nature de ces requêtes.

(10) 2. Le coin XHTML

- (a) Un certain nombre de problèmes sont présents dans ce fichier XHTML qui rendent sa validation impossible. Indiquez ces erreurs en mentionnant le numéro de ligne les concernant, le motif de l'erreur et la façon de corriger l'erreur<sup>1</sup>.

---

<sup>1</sup>Supprimer un élément n'est pas une solution qui vous sera créditée. Les numéros de lignes ne font bien sûr pas parti du code XHTML...

```
1 - <xml version=1.0>
2 - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 - "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 -
5 - <html xmlns="http://www.w3.org/1999/xhtml">
6 -     <head>
7 -         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
8 -         <link rel="stylesheet" type="text/css" href="myshop.css"/>
9 -     </head>
10-
11-    <body>
12-        Bienvenue au Magasin Povigros<br>
13-        <span> Liste des articles en commande:</span>
14-        <dl>
15-            <dt ID=art1>Article 1</dt>
16-            <dd>25</dd>
17-            <dt class="red">
18-                <a src="http://www.myshop.com/articles2.html">Article 2</dt>
19-            <dd>250</dd>
20-
21-            <dt></dt>
22-            <dd class="red">18</dd>
23-        </dl>
24-
25-        <p>Les prix sont donnés en dollard canadien.
26-        <p>Liste des articles disponibles:
27-        <ul>
28-            <li> Article 1
29-                <ul> <li class="red">1 a </li> <li>1 b</li>
30-                    <li>1 c</li> <li class="green">1 d</li>
31-            <li> Article 2
32-                <ul>
33-                    <li>2 a</li> <li class="red">2 b</li> <li>2 c</li> <li>2 d</li>
34-                </ul>
35-            <li> Article 3
36-            <li> Article 4
37-        </ul>
38-        Consultez nos <a src=http://www.myshop.com/info.html>promotions</a>
39-    </body>
40- </html>
```

- (b) Indiquez les commandes CSS qui permettent:
- de mettre en jaune le texte de l'élément d'identificateur `art1`
  - de mettre en rouge la couleur de fond des éléments qui possèdent la propriété `red`
  - de mettre en gras le texte des éléments `li` qui possèdent l'attribut `class`
  - de mettre en souligné le texte du troisième élément (item) de la deuxième liste `ul` d'un document HTML.
  - d'ajouter la chaîne `ift` devant le texte de tous les éléments (items) de toutes les listes `ul` d'un document HTML, à l'exception du second élément de chaque liste.
  - Que veut dire le sélecteur CSS: `a.x b#y > li + li` ? Donnez une instance où ce sélecteur pourrait s'appliquer en soulignant le ou les éléments sélectionnés.

(12) 3. Schéma XML / DTD

- Indiquez la différence entre un document XML *valide* et un document *bien formé*.
- L'exemple XML qui suit contient 6 éléments différents qui permettent d'encoder des expressions arithmétiques où les opérateurs `plus` et `prod` ont une arité<sup>2</sup> variable. (Voir la question 5c pour une forme mathématique des expressions représentées.)

**program** l'élément racine qui peut prendre un attribut `name` (de type texte) et qui contient une séquence d'au moins une instruction (élément `inst`).

**inst** qui contient optionnellement un commentaire (élément `comment`) suivi soit d'un élément `plus`, soit d'un élément `prod`.

**comment** un élément qui contient seulement du texte.

**plus** est un élément qui contient un nombre quelconque d'entiers (élément `entier`) et d'éléments `plus` ou `prod`.

**prod** est un élément qui contient un nombre quelconque d'entiers (élément `entier`) et d'éléments `plus` ou `prod`.

**entier** est un élément dont l'attribut `val` contient une valeur entière.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<program xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="expression.xsd" name="prog1">
```

```
  <inst>
    <plus><entier val="10" /> <entier val="5" /></plus>
  </inst>
  <inst>
    <comment>2+4+5</comment>
    <plus> <entier val="2" /> <entier val="4" /> <entier val="5" /> </plus>
  </inst>
</inst>
```

---

<sup>2</sup>nombre d'arguments.

```

    <plus>
      <entier val="5" />
      <prod> <entier val="3" /> <entier val="4" /> </prod>
      <entier val="7" />
      <prod>
        <entier val="12" />
        <plus>
          <entier val="1" />
          <entier val="2" />
          <prod> <entier val="2" /> <entier val="2" /> </prod>
          <entier val="3" />
        </plus>
      <entier val="4" />
    </prod>
    <entier val="5" />
  </plus>
</inst>
</program>

```

- Indiquez parmi les 6 éléments du langage ceux qui sont de type simple.
- Donnez le schéma XML du langage défini par les 6 éléments décrits plus haut.
- Donnez le schéma DTD des éléments **program** et **prod**.
- Indiquez les modifications à apporter à votre schéma XML de manière à ce que le contenu de l'attribut **name** de l'élément **program** commence par une majuscule.

(14) 4. Programmation XSLT

- Écrire une instruction XPath qui calcule le nombre d'éléments **entier** contenus dans une instance XML donnée.
- Écrire une feuille de transformation XSLT qui prend en entrée une instance du langage défini à la question 3 et qui affiche en sortie une page HTML contenant la liste des commentaires présents dans l'instance. Vous n'avez pas à vous soucier des entêtes de la page HTML générée par transformation (commencez avec l'élément **<html>** qu'il est inutile de qualifier).
- Écrire une feuille de transformation XSLT qui transforme une instance XML telle que définie à la question 3 en instance XML qui contient le résultat de chaque une page HTML contenant les expressions infixées correspondant aux instructions rencontrées dans l'instance de telle sorte que seules les sous-expressions contenant plus d'un argument soient parenthésées. Pour l'instance donnée en exemple, la page devrait contenir:

10 + 5

2 + 4 + 5

5 + (3 \* 4) + 7 + (12 \* (1 + 2 + (2 \* 2) + 3) \* 4) + 5

Dans cette sortie, 12 n'est par exemple pas parenthésé, car il s'agit d'une sous-expression contenant un seul argument (12).

- (d) Écrire une feuille de transformation qui prend en entrée une instance XML telle que définie à la question 3 et qui produit une instance où toutes les instructions sont calculées. L'instance de la question 3 devrait par exemple être transformée en:

```
<exec>
  <inst> 15 </inst>
  <inst> 11 </inst>
  <inst> 509 </inst>
</exec>
```