

Solutions :

1. Quelles sont les valeurs affichées par cette **clôture** javaScript : (2.00 points)

console.log(x(1)) 5 0.75

console.log(y(2)); 5 0.75

console.log(x(2)); 7 0.5

2. Quelles sont les valeurs affichées par la fonction immédiatement invoquée (IIFE) ?(1,5 points)

console.log(T[i]) 4 9 16

3. Soit books une chaîne JSON définit comme suite : (1.5 points)

a. obj2 = JSON.parse(books); 0.5

b. obj2.books[1].title 1.00

4. Comment savoir si une requête **ajax** est terminée ou non ? (1.00 point)

si la propriété **ReadyState** est égale à 4 1.00

5. Soit le script **PHP** suivant :

- a. (1.5 points)

```
$insert = "INSERT INTO ville (N_ville, code_Post) 0.5
VALUES (:name, :code)";
$stmt = $file_db->prepare($insert);
$stmt->bindParam(':name', $name); 0.5
$stmt->bindParam(':code', $code);
$name="setif"; $code="19000";
$stmt->execute();
$name="alger"; $code="16000"; 0.5
$stmt->execute();
$name="tlemcen"; $code="13000";
$stmt->execute();
```

- b. (1.5 points)

```
if(isset($_GET['code_ville'])){ 0.25
$codeV = $_GET['code_ville'];
}
$stmt = $file_db->prepare("SELECT * FROM ville WHERE code_Post=:id"); 0.25
$stmt->bindValue(':id', $codeV, PDO::PARAM_INT);0.25
$stmt->execute();0.25
$rows = $stmt->fetch(PDO::FETCH_ASSOC); 0.25
echo json_encode($rows); 0.5
```

c. (2.00 points)

```
function chercher() {  
    var villeC = 'code_ville=' + $("#Mycode").val(); 0.25  
    var ajaxCall=$.ajax({ 0.25  
        type : 'GET',  
        url: 'examen.php', 0.25  
        dataType: 'json', 0.25  
        data : villeC, 0.25  
    });  
    ajaxCall.done(function (data) { 0.25  
        $("#divVille").html(  
            '<p>'+ +data.N_ville+ 0.25  
            '</p>'  
        )  
    });  
    ajaxCall.fail(function() {  
        $("#divVille").html("error!"); 0.25  
    });  
}
```

6. Soit le document XML suivant

a. Écrivez une DTD qui valide ce document (la DTD doit être la plus stricte.) (2.00 points)

```
<!ELEMENT etudiants (etudiant+)> 0.25  
<!ATTLIST etudiant id ID #REQUIRED sex (M|F) "M"> 0.75  
<!ELEMENT etudiant (nom,prenom,moy)> 0.25  
<!ELEMENT nom (#PCDATA)>  
<!ELEMENT prenom (#PCDATA)> 0.75  
<!ELEMENT moy (#PCDATA)>
```

b. Donnez les expressions XPath permettant de sélectionner : (3.00 points)

- i. `count(//etudiant[moy>=10])` 1.00
- ii. `//etudiant[moy>12.5]/nom` 1.00
- iii. `count(//etudiant[(moy>=10) and (@sex="F")])` 1.00

c. (4.00 points)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"                                0.5
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  <xsl:template match="etudiants">0.25
    <h1>Liste des étudiants</h1>
    <br/>
    <table border="1">
      <tr>
        <th>ID</th>
        <th>Nom</th>
        <th>Prénom</th> 0.5
        <th>Admis</th>
      </tr>
      <xsl:apply-templates select="//etudiant" /> 0.5
    </table>
  </xsl:template>
  <xsl:template match="//etudiant"> 0.5
    <tr>
      <td> <xsl:value-of select="@id" /></td> 0.25
      <td> <xsl:value-of select="nom" /></td> 0.25
      <td> <xsl:value-of select="prenom" /></td> 0.25
      <xsl:if test="moy >= 10"> 0.5
        <td> Oui </td>
      </xsl:if>
      <xsl:if test="moy < 10"> 0.5
        <td> Non </td>
      </xsl:if>
    </tr>
  </xsl:template>
</xsl:stylesheet>
```