

# **ANNALES 2018-2019**

## **SEMESTRES 3 & 4**

### **SESSION 1 & 2**

#### **UEs INFORMATIQUE :**

- **PROGRAMMATION FONCTIONNELLE**
- **ARCHITECTURE DES ORDINATEURS 2**
- **BASE DE DONNEES RELATIONNELLE**
- **PROGRAMMATION WEB**
- **PROGRAMMATION OBJET 1**

## PROGRAMMATION FONCTIONNELLE

1ère session

**Durée de l'épreuve** : 2 heures

**Documents autorisés** : Cours, travaux dirigés et travaux pratiques

**Attention** : Toute fonction, expression, ... doit être écrite en langage Caml et accompagnée de son **type**.

### Exercice 1 (4 points)

La suite de Syracuse  $s_n(e)$  d'un entier positif  $e$  est définie par récurrence

$$\begin{aligned}
 S_0(e) &= e \\
 S_n(e) &= \begin{cases} S_{n-1}(e) / 2 & \text{si } S_{n-1}(e) \text{ est pair} \\ 3 \times S_{n-1}(e) + 1 & \text{sinon} \end{cases} \quad \text{pour } n \geq 1
 \end{aligned}$$

Quelque soit la valeur  $e$ , au bout d'un certain rang, la suite renvoie les trois valeurs successives 4, 2, et 1 indéfiniment.

Par exemple

$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow \dots$   
 $13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \dots$   
 $21 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \dots$

On appelle « temps de vol » d'un nombre  $e$  le rang pour lequel la suite de Syracuse de ce nombre atteint pour la première fois soit 4, soit 2, soit 1.

Sur les exemples ci-dessus, le temps de vol de 3 est 5, celui de 13 est 7 et celui de 21 est 5.

On appelle « altitude maximale » d'un nombre  $e$  la valeur maximale atteinte par la suite de Syracuse de ce nombre.

Sur les exemples ci-dessus, l'altitude maximale de 3 est 16, celle de 13 est 40 et celle de 21 est 64.

#### Questions

1. Ecrire une fonction *temps\_vol* qui étant donné un entier  $e$  renvoie le temps de vol de  $e$
2. Ecrire une fonction *alt\_max* qui étant donné un entier  $e$  renvoie l'altitude maximale de  $e$

Remarque : on pourra utiliser la fonction prédéfinie *max*.

### Exercice 2 (5 points)

Un nombre entier positif est représenté par une liste d'entiers.

Par exemple 359 sera représenté par la liste [ 9 ; 5 ; 3 ]

#### Questions

1. Étant donné deux nombres représentés par des listes, écrire une fonction à deux arguments *addition* qui renvoie l'addition de ces deux nombres.

#### Exemple

L'addition de [9; 9; 9; 9; 9; 9] et [ 2; 6; 1] donnera [1; 6; 1; 0; 0; 0; 1]

(999999 + 162 = 1000161)

Remarque : On pourra écrire plusieurs fonctions pour répondre à la question.

2. Étant donné un chiffre et un nombre représenté par une liste, écrire une fonction à deux arguments *mult\_chiffre* qui renvoie la multiplication de nombre par chiffre.

Exemple

Avec les données 2 et [9; 9; 9], le résultat sera [8; 9; 9; 1]

Remarque : On pourra écrire plusieurs fonctions pour répondre à la question.

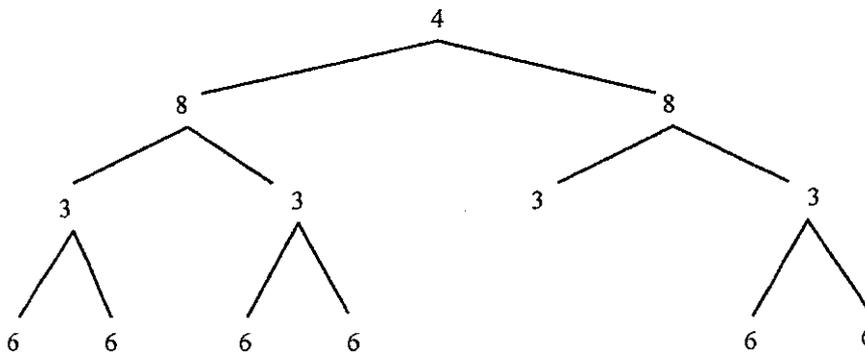
**Exercice 3 (10 points)**

Nous considérons le type suivant :

type 'a arbre\_bin = Vide | Noeud of 'a \* 'a arbre\_bin \* 'a arbre\_bin;;

Un arbre binaire est dit nivelé si et seulement si tous les éléments de même niveau sont égaux.

L'arbre Niv suivant est nivelé



**Questions**

1. Ecrire une fonction *liste\_niveaux* à deux arguments A et n, qui renvoie la liste des éléments des nœuds du niveau n de l'arbre binaire A.
2. Ecrire une fonction *parcours\_largeur* qui étant donné un arbre binaire A renvoie une liste dont les éléments sont les nœuds de l'arbre A parcouru niveau par niveau  
Exemple : Avec l'arbre Niv, on obtient la liste [ 4 ; 8 ; 8 ; 3 ; 3 ; 3 ; 3 ; 6 ; 6 ; 6 ; 6 ; 6 ; 6 ; 6 ]

Remarques :

- Vous pouvez utiliser la fonction *liste\_niveaux* définie à la question 1 et la fonction *hauteur* définie ci-dessous.  

```

#let rec hauteur = fonction
  Vide -> -1
  | Noeud ( _ , sag ,sad) -> 1 + max (hauteur (sag)) (hauteur (sad)) ;;

```
  - Vous pouvez écrire plusieurs fonctions pour répondre à la question
3. Modifier la fonction précédente pour obtenir une liste de liste, chaque « sous liste » contenant les nœuds d'un niveau. Cette nouvelle fonction s'appellera *parcours\_largeur\_niv*.  
Exemple : Avec l'arbre Niv, on obtient la liste  
[[4]; [8; 8]; [3; 3; 3; 3]; [6; 6; 6; 6; 6; 6; 6; 6]]
- Remarque : Vous pouvez écrire plusieurs fonctions pour répondre à la question.

4. Etant donné une liste lst, écrire une fonction *tous\_egaux*, qui teste si tous les éléments de lst sont égaux

5. En utilisant la fonction précédente `tous_egaux` et la fonction `parcours_largeur_niv`, écrire une fonction `test_nivele` qui teste si un arbre binaire est nivelé

Remarque : Vous pouvez écrire plusieurs fonctions pour répondre à la question.

### Exercice 4 (2 points)

La fonction suivante

```
#let exo4 = fun x y z -> x ( y z );;
```

a pour type

```
exo4: ('a -> 'b) -> ('c -> 'a) -> 'c -> 'b = <fun>
```

En déduire le type des fonctions suivantes

1. `let exo4_1 = fun x y z -> x ( y z ) + 2;;`
2. `let exo4_2 = fun x y z -> x ( y 2 ) + z;;`
3. `let exo4_3 = fun x y z -> x ( y 2 ) + 2;;`
4. `let exo4_4 = fun x y z -> x ( ( y 2 ) + 2 ) + 2;;`

## Examen d'Architecture des Ordinateurs 2 du 20/12/2019

**Durée : 2h****Aucun document n'est autorisé.****La calculatrice n'est pas autorisée.****Exercice 1:**

On considère le nombre 211 écrit en décimal. Donnez son écriture dans la base 6 (vous détaillerez la démarche pour obtenir le résultat).

**Exercice 2 :**

Dans une application, des nombres fractionnaires sont stockés sur 16 bits en suivant la représentation en virgule fixe au format [9,7].

- 1) Rappelez le principe de cette représentation.
- 2) Donnez en hexadécimal les 16 bits correspondants au codage de -102,25 (vous commenterez la démarche pour obtenir ce résultat).

**Exercice 3 :**

On considère la portion de mémoire suivante :

25	63	97	61	A5	9A	1E	3F	E6	8C	9A	54	9D	61
\$4E90	\$4E91	\$4E92	\$4E93	\$4E94	\$4E95	\$4E96	\$4E97	\$4E98	\$4E99	\$4E9A	\$4E9B	\$4E9C	\$4E9D

On suppose que tous les registres (de données et d'adresse) contiennent la valeur hexadécimale ABCDEFFF.

Donner les contenus des registres ou zones mémoires modifiés après l'exécution de chacune des 12 lignes suivantes :

- |                     |                      |
|---------------------|----------------------|
| 1. MOVE #100,D0     | 7. MOVE D1,A2        |
| 2. MOVE #2F88,D1    | 8. MOVE.B (A2),D0    |
| 3. MOVE #2F93,A0    | 9. MOVE.B #78,\$4E94 |
| 4. MOVE \$4E96,A1   | 10. MOVE \$4E94,A3   |
| 5. MOVE.B \$4E92,D1 | 11. MOVE (A3)+,D3    |
| 6. ADD.L #1F00,D1   | 12. MOVE 2(A2),D1    |

**Exercice 4 :**

Ecrire le programme assembleur permettant l'affichage d'une chaîne de caractères terminée par 0. On suppose que la fonction AFFCAR existe déjà.

### **Exercice 5 :**

On considère les déclarations et le sous-programme suivant :

```
ORG $2000
VAR : DC.B 72,69,76,76,79,10,13,65,10,13,84,79,85,83
SSPRG:
    MOVE.L    #VAR,A0
    MOVE.L    #14,D7
SSPRG1:
    MOVE.B    (A0)+,D1
    JSR      AFFCAR
    SUB.B     #1,D7
    CMP.B     #0,D7
    BNE      SSPRG1
    RTS
```

- 1) Que se passe t-il si on exécute le sous-programme SSPRG ?
- 2) Quel est le contenu de A0 après l'appel de SSPRG ?
- 3) Ecrire le programme complet en plaçant les éléments au bon endroit. On supposera que le fichier BIBLIO.X68 se situe dans le même dossier que notre fichier.

Rappel : le code ASCII de 'A' en décimal est 65.

Session de rattrapage de Base de Données Relationnelle

2019

---

*Durée 2H00 – Documents non autorisés. Éteindre les téléphones. Barème donné à titre indicatif. Le sujet comporte 3 pages.*

**EXERCICE 1 (4 Points)**

Le club de football de votre village a décidé d'utiliser une base de données afin de garder une trace de tous les matchs joués dans les différents championnats (intervillages, régional, national, amateur, etc.). Les équipes participant aux matchs ont comme caractéristique un nom, un maillot et un stade. Un match est la rencontre de deux équipes à une certaine date donnant lieu à un score. On distingue les matchs « allés » des matchs « retours ». Il est possible aussi de stocker la vidéo du match afin de faire un débriefing avec les entraîneurs. Enfin, les équipes participent donc à des compétitions dont ces matchs font partie.

Établir le modèle étendu Entités/Associations à partir des données présentes dans l'énoncé sans en ajouter. Pour chaque entité, donner la liste des clés candidates et la clé primaire choisie. Justifiez brièvement vos cardinalités et toute autre chose que vous jugeriez nécessaire. Transformer votre modèle dans le modèle relationnel (**attention : vous utiliserez les conventions du cours/TD**).

**EXERCICE 2 (6 Points)**

L'hôpital de Boves décide d'informatiser son système d'information. L'hôpital comprend des patients, des infirmiers, des médecins et des administratifs. Tous comportent un identifiant, un nom, un prénom, un salaire et une date d'embauche. Pour les infirmiers, la date d'obtention de leur diplôme est connue. Pour les médecins la date et l'université d'obtention de leur diplôme sont connus ; ils ont également une ou plusieurs spécialités. On connaît le niveau de diplôme (BEPC, BAC, BAC+2, etc.) des administratifs. Pour les patients on a la première date de passage à l'hôpital.

Les patients sont dans des chambres individuelles dont on connaît le numéro et le service. Les services sont caractérisés par leur nom et un responsable qui est un médecin. Les services comprennent des infirmiers et des médecins. Ceux-ci ne font partie que d'un seul service. Les médecins et infirmiers ont en charge les patients. Les infirmiers en chef dirigent un ensemble d'infirmiers. Il n'y a qu'un infirmier en chef par service et ils dirigent des infirmiers appartenant au même service qu'eux.

Les opérations sont réalisées par un médecin qui possède la spécialité dont l'opération relève. Lors des opérations, les médecins sont aidés par des infirmiers.

1. Établir le Modèle **Étendu** de Conception des Données. Donnez le modèle relationnel correspondant. (attention : vous utiliserez les conventions du cours/TD).
2. Si l'on suppose qu'une chambre contient deux lits, quelles sont les modifications à apporter à votre MCD ?

### EXERCICE 3 (7 Points)

Soient les relations :

Marque( **IDM**, NomM)

Voiture( **Plaque**, Modele, couleur, IDM, IDP)

Proprietaire( **IDP**, NomP, PrenomP, AdresseP)

Mecanicien( **IDMec**, NomMec, PrenomMec)

Repare( **IDMec**, **Plaque**, **Date**, Montant) .

Les clés primaires sont en **gras**, les clés étrangères sont soulignées.

1. Donnez le Modèle de Conception des Données correspondant
2. Exprimez ces questions en opération sur les relations sous la forme de schéma algébrique (au choix linéaire ou arborescente) et/ou sous forme SQL (suivant l'énoncé) :
  - a) Quel est le chiffre d'affaire du garage ? (SQL).
  - b) A quel nom de propriétaire appartient la voiture dont la réparation est la plus chère ? (SQL)
  - c) Pour chaque nom de marque de voiture, donnez le montant total des réparations (SQL)
  - d) Quel est le mécanicien (IDMec) qui a travaillé sur la voiture de Jean et la voiture de Paul ? (algébrique et SQL).
  - e) Afficher les noms des mécaniciens qui ont fait des réparations sur les voitures de marques « Renault » ou « Peugeot » (algébrique et SQL).
  - f) Quels sont les noms des mécaniciens qui ont réparé une voiture dont la plaque est plus grande que la plaque de la voiture de « Fred » (on supposera que Fred n'a qu'une voiture et que les plaques sont des entiers) (algébrique) ?
  - g) Écrire une vue Vue\_Meca\_Montant(IDMec, Montant) qui donne les marques des véhicules réparés par IDMec et dont le montant de la réparation est inférieur à Montant.
  - h) Énoncer une requête dont la réponse est une division. Donner la requête , les 3 formes SQL et la forme algébrique pour y répondre.

**EXERCICE 4 (3 Points)**

Soient les deux relations R et T :

R	A	B	C
	1	2	5
	1	1	3
	1	3	5
	2	1	3
	3	2	5
	3	4	5
	4	3	5
	4	1	5
	4	2	5
	5	3	5

T	B	C	D
	1	3	3
	2	5	5
	3	5	4

Que retourne  $\frac{\sigma_{A \geq 3}(R)}{\Pi_{B,C}(\sigma_{B > 1}(T))}$  ?

# Examen de Programmation Web

## Documents autorisés : notes de cours et TP.

(Le barême est donné à titre indicatif et est susceptible de subir des modifications)

### **Exercice 1.**(5 points)

Pour chaque question ci-dessous, indiquez la ou les lettres des réponses correctes correspondantes. Attention, pour chaque question, une lettre manquante ou incorrecte entraîne 0 à la question.

**Question 1 :** Pour quelles valeurs de \$foo ce script affichera t'il "hello" ?

```
<?php
if(preg_match('/a{2,}b[0-9]*c?de/', $foo))
    print 'hello';
?>
```

- a) aabde
- b) aab123456de
- c) abcde
- d) aabcde
- e) aa123cde

**Question 2 :** Quelle sera la sortie du script suivant ?

```
<?php
$foo = 0 || 15;
var_dump($foo);
echo $foo;
?>
```

- a) une erreur
- b) 0
- c) 1
- d) 15
- e) true

**Question 3 :** Quelle sera la sortie du script suivant ?

```
<?php
$a = array("A" => "A", "C"=> NULL, 78909 => "D");
echo count($a);
?>
```

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4
- f) 5
- g) 6
- h) Une erreur

**Question 4 :** Quelle sera la sortie du script suivant ?

```
<?php
$x = function($a, $b, $c){
    return "$a|$b|$c";
}
print $x(1,2,3);
?>
```

- a) 0
- b) 1
- c) 3|2|1
- d) 1|2|3
- e) Une erreur
- f) Aucune de ces réponses

**Question 5 :** Quelle sera la sortie du script suivant ?

```
<?php
$foo = "08"; $bar = 08; $baz = " 47";
$foo += 15; $bar += "15"; $baz += 11;
print $foo.$bar.$baz;
?>
```

- a) Une erreur
- b) 232358
- c) 231558
- d) 151511
- e) 152311
- f) 152358
- g) 231547

## **Exercice 2.(6 points)**

Pour chaque question une réponse erronée ou partielle entraîne 0 à la question.

**Question 1 :** Indiquez ce qu'affiche en sortie le code suivant.

```
<?php
for($i = 0; $i < 1.02; $i += 0.17){
    $a[$i] = $i;
}
print count($a);
?>
```

**Question 2 :** Si le code suivant provoque une erreur indiquez « erreur » en réponse à cette question. Sinon, indiquez ce que ce code affiche en sortie.

```
<?php
function maFonction(&$param1){
    $param1 = ++$param1;
}
$struc="45"+2;
maFonction($truc);
echo $truc, $truc*2;
?>
```

**Question 3 :** Indiquez ce qu'affiche en sortie le code suivant.

```
<?php
print strlen('o\n2') * strlen('\n');
?>
```

### Exercice 3.(9 points)

Ci-dessous se trouve le script « affiche\_tout.php » de la table dictionnaire de la base de données dupont (user : dupont, mot de passe : herge, serveur : localhost) :

```
<html>
<body>
<h2 align=center> contenu du dictionnaire</h2>
<p>
nom, definition :<br>
<?php
$link = mysqli_connect('localhost', 'dupont', 'herge', 'dupont') or die("erreur de connexion");
$result = mysqli_query($link, "SELECT mot, definition
FROM dictionnaire ") or die("erreur de connexion");
mysqli_close($link);
while ($row = mysqli_fetch_array($result))
    echo $row[0] . ' : ' . $row[1] . '<br>';
?>
</p>
</body>
</html>
```

**Question 1 :** On souhaite effectuer l'insertion de mot dans le dictionnaire grâce au fichier *ajout.html* ci-dessous.

```
<html>
<body>
<h2 align=center> Ajouter une d&eacute;finition dans le dictionnaire
</h2>
<form action="ajout.php" method="POST">
mot :<input type="text" name="mot"><br>
d&eacute;finition :
<textarea name="definition" cols="40" rows="6"></textarea><br>
<input type="submit" name='submit' value="ajouter" >
</form>
</body>
</html>
```

Ecrire le fichier *ajout.php* qui ajoute le mot et sa définition (renseigné via le formulaire *ajout.html*) et qui affiche ensuite le contenu du dictionnaire ainsi modifié.

**Question 2 :** Ecrire le code manquant au script *supprime.php* donné ci-dessous qui permet de supprimer des mots du dictionnaire.

```
<?php
$link = mysqli_connect('localhost','dupont','herge', 'dupont')
    or die ("erreur de connexion a la base de donnees");
if (isset($_POST['submit'])) {
    // code manquant
?>
<html>
<body><?php echo $affiche; ?>
<h2 align=center> suppression de d&eacutefinition dans le dictionnaire </h2>
<table border=1>
<tr>
<th>mot</th><th>d&eacutefinition</th>
</tr>
<?php
    while ( $row = mysqli_fetch_array($result))
        echo '<tr><td><form action="'.$_SERVER['PHP_SELF']
            .'" method="POST">Supprimer <input type="submit" name="submit"
value="'
            . $row[0].'"></form></td><td>'. $row[1].'"</td></tr>';
?>
</table>
</body>
</html>
```

# Licence Informatique 2ème année

## Programmation Objet 1

Examen de seconde session – 28/06/2019 – 2h

Les notes de cours, TD et TP sont autorisées.

**Exercice 1** (7 points) : on veut simuler la diffusion des fausses nouvelles sur un réseau social. Une nouvelle est constituée par un texte et un niveau de fiabilité (entre 0 et 100%).

a- Écrire une classe pour représenter les nouvelles (1 point).

Le réseau social est constitué de personnes qui ont un nom, des personnes amies, et la liste des nouvelles qu'elles publient sur leur page personnelle. On doit pouvoir ajouter un ami à une personne, et chaque personne peut recevoir une nouvelle d'une autre personne. Mais la façon dont elle traite la nouvelle reçue peut changer d'une personne à l'autre.

b- Écrire une classe pour représenter les personnes (3 points).

Quand une personne reçoit une nouvelle, elle peut l'ignorer, la publier (la nouvelle est alors ajoutée à sa liste de nouvelles publiées sur sa page), la transmettre à tous ses amis (qui donc reçoivent cette nouvelle). Certaines personnes sont crédules et, quand elles reçoivent une nouvelle qu'elles n'ont pas encore sur leur page, elles la publient sur leur page et la transmettent à tous leurs amis. Certaines personnes sont méfiantes et ne publient et transmettent que les nouvelles reçues d'une personne amie.

c- Proposer du code objet pour représenter les personnes crédules et les personnes méfiantes (3 points).

**Exercice 2** (6 points) : un acronyme est un mot formé par les initiales d'autres mots (ex.: ENA est l'acronyme de "Ecole Nationale d'Administration"). *On suppose ici que les acronymes sont constitués par les premières lettres des mots qui les composent, même si ce n'est pas tout à fait vrai (ex.: l'acronyme de Ecole Nationale d'Administration devrait être ENDA).* Les acronymes sont des listes de mots et doivent obligatoirement implémenter l'interface suivante :

```
public interface Ecrivable{
    public void ecrire(); // écrit l'objet à l'écran
}
```

En outre, les acronymes doivent avoir une méthode `String getSigle()` qui renvoie la chaîne de caractères constituée par les premières lettres de chacun des mots (NB : la méthode `substring(int i, int j)` de la classe `String` renvoie la chaîne constituée par les caractères situés entre les indices `i` et `j-1` inclus).

a- Écrire une classe `Acronyme` pour représenter les acronymes (2 points).

Les acronymes imbriqués sont des acronymes qui peuvent contenir des mots, mais aussi des objets de type `Acronyme` (ex.: AJAX est l'acronyme de "Asynchronous JavaScript And XML", XML étant l'acronyme de "eXtensible Markup Language").

b- Proposer du code objet pour représenter les acronymes imbriqués, qui doivent être des sous-types d'acronymes et avoir la méthode `getSigle` (4 points).

**Exercice 3** (6 points) : on veut réaliser des interfaces graphiques pour traduire des phrases. Les objets chargés de traduire les phrases implémentent tous l'interface `Traducteur` :

```
public interface Traducteur{  
    public String traduit(String s); // renvoie la traduction de s  
}
```

L'interface graphique doit ressembler à l'image suivante :



a- Écrire une classe qui représente les interfaces graphiques de traduction, de façon à ce que quand on appuie sur le bouton "Traduire", on affiche dans la zone de texte de droite la traduction de la phrase écrite dans la zone de texte de gauche (5 points).

On suppose qu'il existe une classe `FR2GB` qui implémente l'interface `Traducteur` et possède un constructeur sans paramètre. La méthode `traduit` de la classe `FR2GB` traduit la chaîne paramètre `s` du français à l'anglais.

b- Écrire une fonction `main` qui crée une interface permettant de traduire des phrases du français à l'anglais (1 point).