

ANNALES 2018-2019

SEMESTRES 5 & 6

SESSION 1 & 2

UEs INFORMATIQUE :

- **LANGAGES FORMELS**
- **THEORIE DE SYSTEME D'EXPLOITATION 1**
- **MODELISATION OBJET**
- **THEORIE DE SYSTEME D'EXPLOITATION 2**
- **PROGRAMMATION OBJET 2**
- **METHODES D'ANALYSE ET DE CONCEPTION DES SI**
- **FRAMEWORK**
- **METHODES QUANTITATIVES ET DECISION**
- **XML**
- **SYSTEME INFO COMPTABLE**
- **RESEAU**
- **INTELLIGENCE ARTIFICIELLE**

**Examen de «Langages Formels» du premier semestre
Mercredi 19 décembre 2018 – Durée: 2 heures**

- Aucun document, calculatrices, portables et autres machines ne sont pas autorisés.
- Toutes les réponses doivent être justifiées - il sera tenu compte de la qualité de la rédaction.
- *Le barème entre parenthèses est indicatif*

Exercice 1 (4 points)

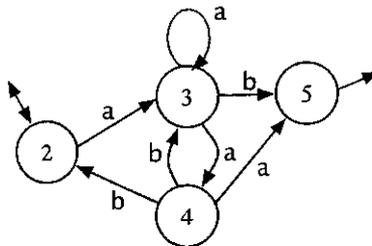
1. Donner la définition d'un automate minimal.
2. Soit Aut , l'automate reconnaissant le langage L :

$$Aut = \langle \{a, b, c, d\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \{1, 2, 5, 7\}, \{3, 5, 7, 9, 10\}, \{(1, a, 3), (1, b, 4), (2, b, 3), (3, b, 1), (5, c, 8), (2, a, 6), (4, d, 10), (2, b, 7), (5, a, 8), (3, c, 5)\} \rangle$$

Donner l'automate Aut_c reconnaissant le langage $L_2 = A^* \setminus L$

3. Énoncer le lemme de l'étoile.
4. Donner les deux premiers langages (dans le sens de l'inclusion) de la hiérarchie de Chomsky en expliquant brièvement la reconnaissabilité des langages impliqués.

Exercice 2 (3 points)



1. En utilisant la méthode par élimination d'états, calculez l'expression rationnelle du langage reconnu par l'automate ci-dessus.

Exercice 3 (3 points) Soit l'automate suivant reconnaissant le langage L :

$$Aut = \langle \{a, b\}, \{1, 2, 3, 4, 5\}, \{1\}, \{5\}, \{(1, a, 2), (2, a, 3), (3, b, 4), (4, b, 4), (4, a, 5)\} \rangle$$

1. En utilisant la méthode par système d'équation, calculez l'expression rationnelle du langage reconnu par l'automate Aut .

Exercice 4 (5 points) Soit l'automate suivant reconnaissant le langage L :

$$Aut = \langle \{a, b\}, \{1, 2, 3, 4, 5, 6\}, \{1\}, \{3, 5, 6\}, \delta \rangle$$

avec $\delta = \{(1, a, 2), (1, b, 2), (2, b, 3), (3, a, 2), (1, b, 4), (4, b, 4), (4, b, 5), (1, b, 5), (5, a, 6), (6, a, 5)\}$

1. En utilisant l'algorithme de calcul de l'automate déterministe complet, donner la table de transition associé à A_{ut} .
2. À partir de la table de transition obtenu précédemment, donner l'automate déterministe complet équivalent à A_{ut} sous la forme d'un 5-uplet.

Exercice 5 (5 points)

Calculer les résiduels des langages $a^* + b^*$ et $A^* \setminus (A^*abA^* + A^*baA^*)$. Que peut-on en conclure ?

UNIVERSITÉ DE PICARDIE - JULES VERNE

UFR des Sciences

L3 INFO & MIAGE: Examen de la 2^{ème} Session Juin 2019

Théorie de Système d'Exploitation 1 : Durée: 2 heures

(Documents non autorisés)

Problème 1

On considère un système de huit tâches $S_2 = \{T_1; T_2, T_3; T_4; T_5; T_6; T_7; T_8\}$ telles que:

T_1 précède T_4 et T_5 ; T_2 précède T_4 et T_5 ; T_3 précède T_6 ; T_4 précède T_6 ; T_5 précède T_8 ;

T_6 précède T_7 et T_7 précède T_8 .

1. Donner le graphe de précédence associé à ce système de tâches. (1pt)
2. Donner le programme parallèle associé à ce système en utilisant la structure Parbegin/Parent. (2pts)

Problème 2 (4pts)

Un système comprend deux ressources, avec chacune deux modalités d'accès a et b. Deux processus s'exécutant en parallèle peuvent simultanément accéder à une ressource suivant la modalité d'accès a, mais la modalité d'accès b doit être utilisée en exclusion mutuelle. Indiquer une exécution conduisant à un blocage.

Problème 3

Une fois que le blocage est détecté, le système doit prendre des mesures pour le supprimer et permettre la reprise de la progression des processus bloqués. Les solutions prévoient la réquisition des ressources de l'ensemble ou d'une partie des processus bloqués. Un problème se pose: quels processus choisir comme victimes d'une réquisition ?

Certains systèmes choisissent de réquisitionner les ressources des processus les uns après les autres jusqu'à disparition du blocage.

1. Quels sont les inconvénients de ce choix? (2pts)

Dans d'autres systèmes, le choix des victimes dans l'ensemble des processus bloqués peut être déterminé par la priorité des processus.

2. Quels sont les inconvénients de ce choix? (3pts)
3. Toujours dans le cas où le choix des victimes dans l'ensemble des processus bloqués est déterminé par la priorité des processus, comment peut-on éviter les inconvénients de la question 2 ci-dessus? (2pts)

Problème 4

Une bibliothèque réservée aux étudiants est constituée d'une entrée, d'une salle de lecture et des guichets pour le prêt et remise des livres. La réglementation en matière de sécurité dans les lieux publics définit un nombre maximal EMAX de personnes présentes simultanément dans un espace donné. Pour respecter cette réglementation, il a été mis en place un système de droits d'accès à la bibliothèque disponibles seulement en EMAX exemplaires. Ces droits d'accès sont matérialisés par des cartes magnétiques.

Un étudiant qui arrive doit se procurer une carte magnétique en présentant sa carte d'étudiant ; à la sortie, il rend la carte magnétique. D'autre part pour pouvoir travailler l'étudiant doit aussi acquérir une chaise et un bout de table (on supposera qu'il y a toujours de bouts de table disponibles). Il y a exactement EMAX chaises dans la salle. L'étudiant se procure la carte magnétique et la chaise, l'une après l'autre, dans n'importe quel ordre, et de manière complètement indépendante. Lorsque l'étudiant possède carte et chaise, il peut travailler.

1. Montrer qu'il peut se produire de blocage dans ce système. (4pts)
2. Indiquer comment éviter ce blocage. (2pts)

N.B: Justifier toutes les réponses. Une réponse non justifiée sera considérée comme nulle.

Examen de 2de session 2018-2019

Seuls les polycopiés et notes de cours, de TD et de TP sont autorisés.
Ce sujet comporte en tout quatre parties et 4 pages.

| | |
|--------------------|--------------|
| Date de l'examen : | 17 juin 2019 |
| Temps alloué : | 2 heures |

Partie I - Spécification- étude statique

- 1 Il s'agit de développer un « jeu d'habillage pour enfant ». Dans cette partie, vous allez effectuer la
2 phase de spécification des besoins pour le développement de ce logiciel.
- 3 Vous avez interviewé le client commanditaire de ce jeu. Voici les éléments que vous avez récoltés.
4 Étant donné un thème particulier (plage, neige, forêt, etc.), le but du jeu est que l'enfant trouve une
5 combinaison d'habits qui soient en adéquation avec le thème proposé. Pour cela, l'enfant dispose de
6 plusieurs combinaisons d'habits possibles, parmi lesquelles il doit choisir. Les habits proposés
7 peuvent correspondre au thème ou non.
- 8 Le but du jeu est énoncé oralement lors du premier lancement du logiciel, mais l'enfant pourra
9 demander à réentendre la consigne à n'importe quel moment du jeu.
- 10 Au fur et à mesure du choix de l'enfant, le jeu indique la bonne ou la mauvaise adéquation avec le
11 thème proposé. L'habillage est complet quand l'enfant a choisi un vêtement pour chaque partie du
12 corps : tête, visage, haut du corps, bas du corps, mains et pieds. Lorsque l'habillage est complet et
13 correct, l'enfant peut éventuellement, rejouer avec le même thème ou changer de thème.
- 14 Le logiciel est fourni avec deux thèmes (plage et neige). Un adulte peut ajouter un thème. Ne pas
15 oublier que pour certains thèmes, la présence d'un vêtement sur une partie du corps est nécessaire,
16 optionnelle ou exclue. Voici quelques exemples : le bonnet (casquette), les lunettes de soleil, les
17 chaussures... Ainsi, lors de la création du thème, l'adulte indique le caractère obligatoire (ou autre)
18 de la présence d'un vêtement sur cette partie. Il faut que le thème soit complet afin que le thème
19 apparaisse en tant que choix supplémentaire. Un thème est complet si l'adulte a :
- 20 - ajouté une image qui représente le thème (paysage ou décors)
21 - défini quelles parties du corps devaient être couvertes
22 - fourni au moins une image de vêtement pour les différentes parties du corps à couvrir.
- 23 Il peut aussi réutiliser les images déjà présentes dans le jeu (par exemple, des lunettes de soleil).

Figure 1 correspond à ces fonctionnalités.

Question 1. Commentez la Figure 1. Vous pouvez par exemple écrire

Le... « ajouter un habit » est justifié la ligne numéro...

Pensez à donner la signification des cinq types de traits et à utiliser le jargon UML.

Question 2. Pourquoi n'y a-t-il pas de CU « vérifier habillage » qui vérifie si l'habillage est complet ni de CU « vérifier thème » qui vérifie si un nouveau thème est complet ?

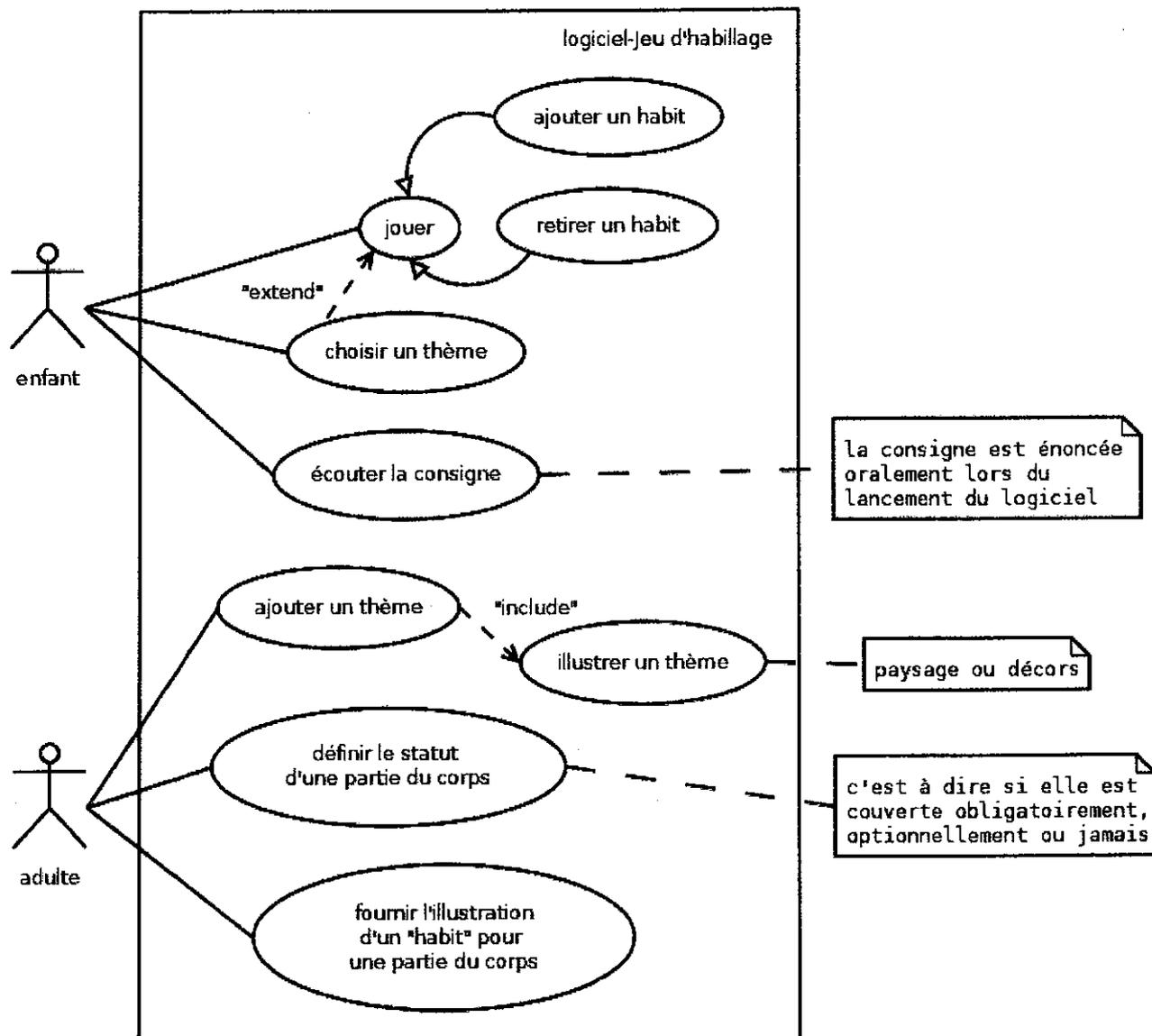


Figure 1. Exemple de DCU pour le jeu d'habillage

Partie II. Spécification - étude dynamique

Question 3. Réalisez l'étude dynamique du CU « *jouer* » avec un diagramme d'activité.

Question 4. Réalisez l'étude dynamique du CU « *choisir thème* » avec un diagramme d'activité.

Question 5. Élaborez un **diagramme de séquences** pour décrire un scénario d'exécution du système où l'enfant -seulement- interagit avec le système et une interface homme-machine, et où vous faites intervenir tous les CU identifiés pour lui.

Question 6. À partir des opérations système que vous aurez identifiées dans la question précédente, déduisez la **classe Système de spécification**.

Partie III. Analyse – étude statique

Au cours des cinq séances de travaux pratiques cette année, nous avons modélisé un logiciel de gestion de projets. Lors de la phase de spécification, nous avons identifié le CU « *affecter un contrôleur de tâche* ». Vous avez aussi identifié l'opération `affecterContrôleurTache(leProjet, laTache, laPersonne1)`.

Le diagramme de la Figure 2 est un sous-ensemble d'un diagramme de classes d'analyse initial.

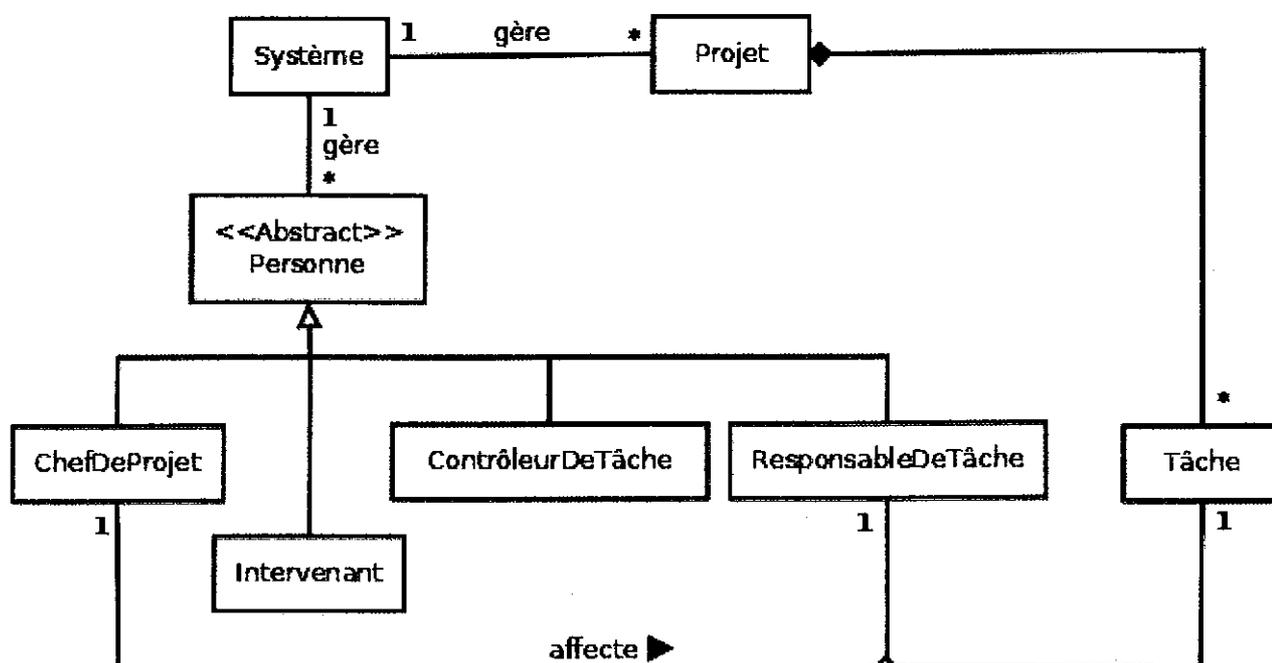


Figure 2. Extrait d'un diagramme de classes d'analyse minimal pour un logiciel de gestion de projets

Question 7. Ce diagramme permet-il de connaître les projets d'un chef de projet ? Justifiez votre réponse.

Question 8. Ce diagramme de classes permet-il à une personne donnée (`toto : Personne`) d'être à la fois le chef de projet d'un projet donné (`p : Projet`) et le contrôleur de la tâche d'une tâche donnée (`t : Tâche`) ? Justifier votre réponse.

Partie IV- Analyse - étude dynamique

En utilisant le diagramme de classes d'analyse de la Figure 3, répondez aux questions suivantes.

Question 9. Pour chaque étiquette numérotée de #1 à #8, dites à quel **type d'attribut** la cardinalité donne lieu lors du passage aux classes de conception. Autrement dit, lorsqu'on supprime le lien, à quel **type d'attribut** est attachée la cardinalité pointée par l'étiquette ? Donnez aussi la classe où elle apparaît.

¹ Pour rappel, en spécification, on ne connaît pas encore le type des paramètres des opérations.

Question 10. Citer les classes qui comportent des **collections de projets**. Expliquer leurs rôles respectifs.

Question 11. Citer les classes qui comportent des **collections de personnes**. Expliquer leurs rôles respectifs.

Question 12. Citer les classes qui comportent des **collections de tâches**. Expliquer leurs rôles respectifs.

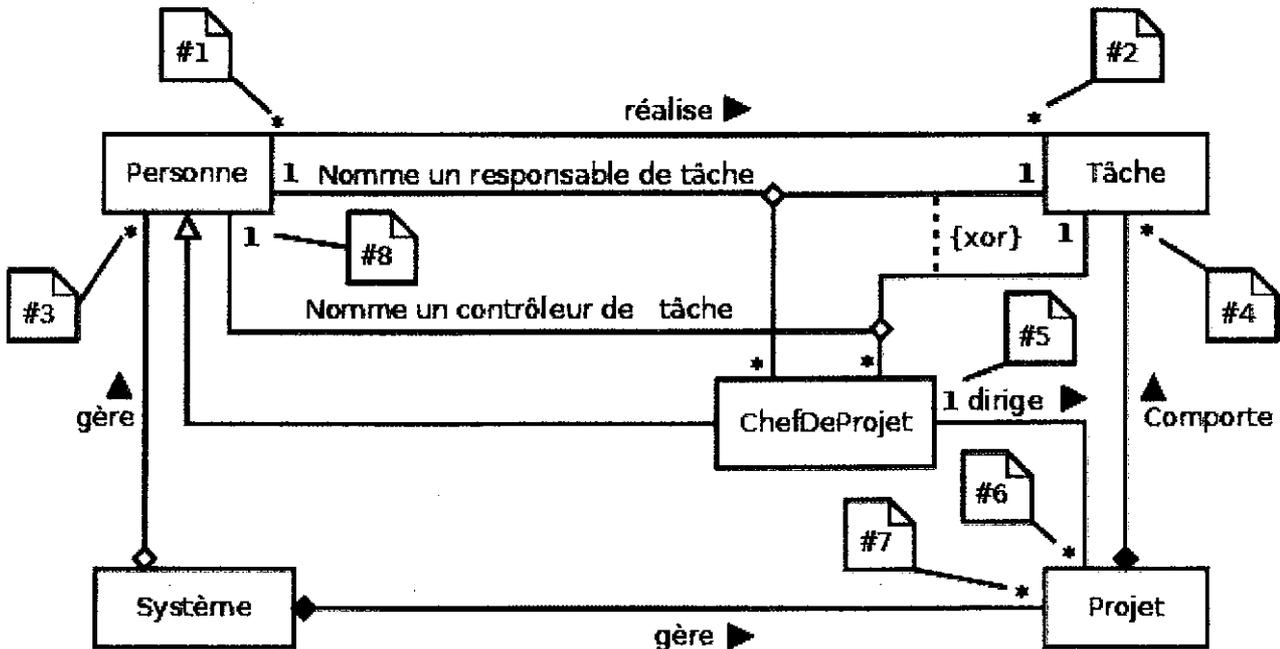


Figure 3. Un diagramme de classe d'analyse minimal pour un logiciel de gestion de projet

Question 13. On rappelle qu'une personne ne peut pas être à la fois contrôleur d'une tâche et responsable de la même tâche. Dessiner le diagramme de séquences d'analyse pour l'opération `affecterContrôleurTache(leProjet, laTache, laPersonne)`. Le paramètre `leProjet` (respectivement `laTache`, `laPersonne`) permet de trouver un projet (resp. une tâche, une personne) dans une collection de projets (resp. de tâches, de personnes).

Partie V- Conception - étude statique

Question 14. Déduisez le diagramme de classes de conception correspondant à la Figure 3. Et ajouter les informations que vous avez déduites de votre réponse à la question 14.

Partie VI- Un diagramme d'objets

Question 15. Proposez un diagramme d'objets correspondant à la Figure 3. Dans ce diagramme, vous ferez figurer deux objets pour chaque classe.

UNIVERSITÉ DE PICARDIE-JULES VERNE

UFR des Sciences

L3 INFO

Examen de Théorie de Système d'Exploitation 2 : 2^{ème} Session

Durée: 2h

(Documents non autorisés)

Problème 1.

1. Expliquer ce qu'est l'équité pour un algorithme d'ordonnement. (1pts)
2. Pourquoi SCAN est-il plus équitable que PCTR? (1pts)
3. Comparer C-SCAN et SCAN.(2pts)

Problème 2

1. Que devient le tourniquet quand le quantum tend vers l'infini ? (2pt)
2. Qu'est-ce qu'un algorithme de pagination stable (2 pt)
3. En allocation non contiguë quel est le comportement de l'algorithme de la seconde chance si toutes les pages ont été référencées dans le précédent intervalle d'horloge ? (2pt)

Problème 3

Dans un système paginé, les pages font 192 mots et la mémoire centrale comprend 3 cadres. On considère les adresses logiques (ou virtuelles) suivantes :

(348), (410), (511), (1160), (977), (824), (1043), (202), (300), (198), (586), (611), (902).

1. Déterminer la suite des pages référencées (1pts)
 2. Calculer les adresses réelles (ou physiques) correspondant à ces adresses virtuelles dans les cas suivants :
 - a. En utilisant LRU comme algorithme de pagination. (2pts)
 - i. Déterminer le taux de défauts de pages. (1 pt)
 - b. En utilisant OPT comme algorithme de pagination. (2pts)
 - i. Déterminer le taux de défauts de pages. (1 pt)
- On suppose que les cadres sont initialement vides.

Problème 4

1. Donner une suite d'adresses logiques, de longueur au moins égale à 6 qui engendre une suite de références de pages telle que, pour le LRU, diviser la taille de la page par deux diminue le nombre de défauts de pages au moins de la moitié. (2pts)
2. Donner un exemple où l'exécution d'une instruction provoque une infinité de défauts de pages (famine) (2pts)

N.B: 1. Présentation et clarté du texte exigées.

2. Justifier toutes les réponses. Une réponse non justifiée sera considérée comme nulle.

Licence Info 3^{ème} année - module Programmation Objet 2

Examen 2^{ème} session – Juin 2019 – durée : 2h00

Seuls documents autorisés : les photocopiés du cours, éventuellement annotés de votre main

Partie 1: Liste des choses à faire ...

Voici la **classe Association** : ne pas modifier !

```
public class Association<T> {
    public Association(String étiquette, T élément) {
        super();
        this.étiquette = étiquette;
        this.élément = élément;
    }
    private String étiquette;
    private T élément;
    public String getEtiquette() {
        return étiquette;
    }
    public T getÉlément() {
        return élément;
    }
    public String toString() {
        return "("+étiquette+"\" : "+élément+"\"";
    }
    public boolean equals(Object autre) {
        if ((autre != null) && autre.getClass().equals(this.getClass())) {
            Association autre2 = (Association)autre ;
            return (this.étiquette.equals(autre2.getEtiquette())
                && this.élément.equals(autre2.getElément()));
        } else
            return false ;
    }
}
```

Voici la **classe Stockage**

Question 1.A : que vous devrez rendre générique

```
import java.util.LinkedList;
import java.util.List;

public class Stockage {
    private List stockage;
    public Stockage() {
        super();
        stockage = new LinkedList();
    }
    public void add(Association association) {
```

```

        if (association != null)
            stockage.add(association);
    }
    public List getElements(String etiquette) {
        List result = new LinkedList();
        for (Association association : stockage)
            if (association.getEtiquette().equals(etiquette))
                result.add(association.getÉlément());
        return result;
    }
    public int remove(Object cetElement) {
        int compteur = 0;
        for (int pos = (stockage.size()-1); pos >= 0; pos--)
            if (stockage.get(pos).getÉlément().equals(cetElement)) {
                stockage.remove(pos);
                compteur++;
            }
        return compteur;
    }
    public String toString() {
        String result = "";
        for (Association association : stockage)
            result += association + " ";
        return result;
    }
}

```

Voici une exécution en console de la classe Todo :

```

f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)
a
ajouter quoi :
ménage
u(rgent), n(ormal), p(eut attendre) ?
p
f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)
a
ajouter quoi :
dentiste
u(rgent), n(ormal), p(eut attendre) ?
u
f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)
a
ajouter quoi :
essence
u(rgent), n(ormal), p(eut attendre) ?
n
f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)
u
urgent(s):

```

dentiste, f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)

t

tout ce qu'il y a à faire:

("Peut_Attendre" : ménage) ("Urgent" : dentiste) ("Normal" : essence)

f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)

a

ajouter quoi :

ménage

u(rgent), n(ormal), p(eut attendre) ?

n

f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)

t

tout ce qu'il y a à faire:

("Peut_Attendre" : ménage) ("Urgent" : dentiste) ("Normal" : essence) ("Normal" : ménage)

f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)

s

supprimer quoi :

ménage

2 occurrence(s) supprimée(s)

f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)

f

Voici la **classe Todo** : ne pas modifier pour la question 1.A !

```
import java.util.Scanner;
```

```
public class Todo {
    private static final String U = "Urgent";
    private static final String N = "Normal";
    private static final String P = "Peut_Attendre";
    public static void main(String args[]) {
        Stockage<String> todo = new Stockage<>();
        Scanner sc = new Scanner(System.in);
        String rep="";
        do {
            System.out.println("f(inir), t(ous les poslts), u(rgent), a(jouter), s(upprimer)");
            rep = sc.next().trim();
            if (rep.equals("t"))
                System.out.println("tout ce qu'il y a à faire:\n "+todo.toString());
            else if (rep.equals("u")) {
                System.out.println("urgent(s):");
                for (String aFaire : todo.getElements(U))
                    System.out.print(aFaire+" ");
            }
            else if (rep.equals("a")) {
                System.out.println("ajouter quoi : ");
                String aFaire = sc.next().trim();
                System.out.println("u(rgent), n(ormal), p(eut attendre) ?");
                String priorité = sc.next().trim();
            }
        }
    }
}
```

```

    if (priorité.equals("u"))
        toDo.add(new Association<String>(U, aFaire));
    else if (priorité.equals("n"))
        toDo.add(new Association<String>(N, aFaire));
    else if (priorité.equals("p"))
        toDo.add(new Association<String>(P, aFaire));
    else
        System.out.println("abandon d'ajouter");
}
else if (rep.equals("s")) {
    System.out.println("supprimer quoi : ");
    String aFaire = sc.next().trim();
    System.out.println(toDo.remove(aFaire) + " occurrence(s) supprimée(s)");
}
}
while (! rep.equals("f")) ;
}
}

```

Question 1.B :

Que faut-il ajouter à la méthode add() de la classe Stockage pour qu'elle génère une exception IllegalArgumentException lorsqu'on tente d'ajouter une association de même étiquette et élément qu'une déjà présente dans le stockage ?

Que faudrait-il ajouter dans la classe Todo ?

Question 1.C :

Que faut-il ajouter à la classe Stockage si plusieurs threads tentent d'y accéder en même temps ?

Partie 2: Todo ... MVC

Voici la **classe ModeleTodo2** : ne pas modifier !

```
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import autrevent.AutreEvent;
import autrevent.AutreEventListener;
import autrevent.AutreEventNotifieur;

public class ModeleTodo2 {
    private AutreEventNotifieur notifieur = new AutreEventNotifieur();

    public static final String U = "Urgent";
    public static final String N = "Normal";
    public static final String P = "Peut_Attendre";
    public static final String[] niveaux = {U, N, P};
    private Map<String,List<String>> todo;

    public ModeleTodo2() {
        todo = new HashMap<>();
        for (String priorité : niveaux) {
            todo.put(priorité, new LinkedList<String>());
        }
    }

    public void add(String aFaire, String priorité) {
        if ( (aFaire != null) && (aFaire.length()>0)
            && isPrioritéValide(priorité)) {
            List<String> liste = todo.get(priorité);
            liste.add(aFaire);
            notifieur.diffuserAutreEvent(new AutreEvent(this,
                new Association<List<String>>(priorité, todo.get(priorité))));
        }
    }

    public static boolean isPrioritéValide(String priorité) {
        return ((priorité !=null) && (priorité.equals(U)
            // priorité.equals(N) // priorité.equals(P)));
    }

    public void del(String aFaire) {
        if ( (aFaire != null) && (aFaire.length()>0)) {
            for (String priorité : niveaux) {
                int compteur=0;
                List<String> liste = todo.get(priorité);
                for (int pos = (liste.size()-1); pos>=0; pos--)
                    if (liste.get(pos).equals(aFaire)) {
                        liste.remove(pos);
                        compteur++;
                    }
                if (compteur>0)
                    notifieur.diffuserAutreEvent(new AutreEvent(this,
                        new Association<List<String>>(priorité, todo.get(priorité))));
            }
        }
    }
}
```

```

    }
}

public void addAutreEventListener(AutreEventListener listener) {
    notifieur.addAutreEventListener(listener);
}

public void removeAutreEventListener(AutreEventListener listener) {
    notifieur.removeAutreEventListener(listener);
}
}

```

Voici la **classe VueTodo2** à compléter :

```

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import autrevent.AutreEvent;
import autrevent.AutreEventListener;
import autrevent.AutreEventNotifieur;

public class VueTodo2 extends Box implements AutreEventListener {
    public static final String ADD = "ajouter";
    public static final String DEL = "supprimer";
    private AutreEventNotifieur notifieur = new AutreEventNotifieur();
    private JTextField aFaire;
    private Map<String, JTextArea> lesListesAffichées;
    private ModeleTodo2 modele;
    public VueTodo2(ModeleTodo2 modele, ControleurTodo2 controleur) {
        super(BoxLayout.Y_AXIS);
        this.modele = modele;
        modele.addAutreEventListener(this);
        notifieur.addAutreEventListener(controleur);
        Box boiteMenu = new Box(BoxLayout.X_AXIS );
        boiteMenu.setBorder(BorderFactory.createTitledBorder("Actions"));
        this.add(boiteMenu);
        boiteMenu.add(new JLabel("à Faire :"));
        aFaire = new JTextField(20);
        boiteMenu.add(aFaire);
    }
}

```

```

boiteMenu.add(new JLabel(" priorité :"));
ButtonGroup prioritéGroupe = new ButtonGroup();
for (String priorité : ModeleTodo.niveaux) {
    JRadioButton radio = new JRadioButton(priorité);
    prioritéGroupe.add(radio);
    boiteMenu.add(radio);
    radio.addActionListener(contrôleur);
}
JButton bouton = new JButton("ajouter");
boiteMenu.add(bouton);
bouton = new JButton("DEL");
boiteMenu.add(bouton);
lesListesAffichées = new HashMap<>();
for (String priorité : ModeleTodo.niveaux) {
    JPanel panel = new JPanel();
    panel.setBorder(BorderFactory.createTitledBorder(priorité));
    JTextArea affichage = new JTextArea(5,20);
    lesListesAffichées.put(priorité, affichage);
    panel.add(affichage);
    this.add(panel);
}
}

public void actionADeclancher(AutreEvent event) {
    if ((event.getDonnee() instanceof Association) && (event.getSource() == modele)) {
        Association<List<String>> paireDonnées = (Association)event.getDonnee();
        String priorité = paireDonnées.getEtiquette();
        List<String> liste = paireDonnées.getÉlément();
        JTextArea listeAffichée = lesListesAffichées.get(priorité);
        listeAffichée.setText("");
        for (String aFaire : liste)
            listeAffichée.append(aFaire + "\n");
    }
}
}
}

```

Voici la **classe ControllerTodo2** à compléter :

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import autrevent.AutreEvent;
import autrevent.AutreEventListener;

public class ControleurTodo2 implements ActionListener {
    private ModeleTodo2 modele;
    private String prioritéCourante = null;
    public ControleurTodo2(ModeleTodo2 modele) {
        this.modele = modele;
    }
    public void actionPerformed(ActionEvent event) {
        String action = event.getActionCommand();
    }
}

```

```
if (ModeleTodo.isPrioritéValide(action))
    prioritéCourante = action;
}
```

Voici une exécution de la classe GuiTodo2 :

| TODO | |
|----------------------|---|
| Actions | |
| à Faire | faire ménage priorité : <input checked="" type="radio"/> Urgent <input type="radio"/> Normal <input type="radio"/> Peut_Attendre <input type="button" value="ajouter"/> <input type="button" value="supprimer"/> |
| Urgent | dentiste : aye! |
| Normal | prendre essence faire ménage |
| Peut_Attendre | faire ménage |

| | | |
|----------------------|---------------------------------|---|
| à Faire | | priorité : <input checked="" type="radio"/> Urgent <input type="radio"/> Normal <input type="radio"/> Peut_Attendre <input type="button" value="ajouter"/> <input type="button" value="supprimer"/> |
| Urgent | dentiste : aye! faire ménage | |
| Normal | prendre essence faire ménage | |
| Peut_Attendre | faire ménage | |

Après demande de suppression de "faire ménage" :

Voici la **classe GuiTodo2** : ne pas modifier !

```
import javax.swing.JFrame;
public class GuiTodo2 extends JFrame {
    public GuiTodo2() {
        super("TODO");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ModeleTodo2 modele = new ModeleTodo2();
        ControleurTodo2 controleur = new ControleurTodo2(modele);
        VueTodo2 vue = new VueTodo2(modele, controleur);
        this.setContentPane(vue);
        this.pack();
        this.setVisible(true);
    }
    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(() -> { new GuiTodo2(); });
    }
}
```

Question 2.A :

Que faut-il ajouter aux boutons ADD et DEL de la VueTodo2 ? Quel code ?

Question 2.B :

Que faut-il ajouter à la classe ControleurTodo2 ? Quel code ?

Université de Picardie Jules Verne – U.F.R. des Sciences
Licence d'Informatique – Troisième année

Examen de «Langages Formels»
Mercredi 19 juin 2019 – Durée: 2 heures

- Aucun document, calculatrices, portables et autres machines ne sont autorisés.
- Toutes les réponses doivent être justifiées - il sera tenu compte de la qualité de la rédaction.
- *Le barème entre parenthèses est indicatif*

Exercice 1 (4 points)

1. Donner la définition formelle d'un automate complet déterministe.
2. Que vaut $|u|$, $|u|_a$ pour les mots u et lettres a suivants ?
 - (a) $u = 0110101$, $a = 0$;
 - (b) $u = 0110101$, $a = 1$;
 - (c) $u = 0110101$, $a = 2$.
3. Quel est le nombre de préfixes d'un mot u en fonction de la longueur de ce mot ?

Exercice 2 (4 points)

1. Dessiner l'automate $\langle A, Q, D, F, \delta \rangle$ où
 - $A = \{a, b, c\}$,
 - $Q = \{1, 2, 3, 4, 5\}$,
 - $D = \{1, 2\}$,
 - $F = \{1, 3\}$,
 - $\delta = \{(1, a, 2), (1, a, 4), (2, a, 3), (3, b, 1), (4, b, 1)\}$

2. Les mots suivants sont-ils reconnus par cet automate : ε , a , b , aa , aba , abb , $abab$.

Exercice 3 (4 points)

Soit l'automate asynchrone $Aut1 = \langle A, Q, D, F, \delta \rangle$ où : $A = \{a, b\}$, $Q = \{1, 2, 3, 4, 5, 6\}$, $D = \{1\}$, $F = \{5\}$ et $\delta = \{(1, a, 2), (1, \varepsilon, 3), (2, \varepsilon, 5), (2, a, 6), (3, b, 2), (4, \varepsilon, 1), (4, \varepsilon, 3), (5, \varepsilon, 1), (5, \varepsilon, 5)\}$.

1. Sans dessiner $Aut1$, construire un automate (sans ε -transitions) $Aut2$ équivalent.

Exercice 4 (4 points)

Déterminer l'automate $\langle \{a, b\}, \{1, 2, 3, 4\}, \{1\}, \{4\}, \{(1, a, 1), (1, a, 2), (1, b, 2), (2, a, 2), (2, a, 3), (2, b, 3), (3, a, 3), (3, a, 4), (3, b, 4), (4, a, 4)\} \rangle$

Exercice 5 (4 points)

Calculer les résiduels du langage $aA^*a \cup bA^*b$ avec $A = \{a, b\}$. Donner l'automate minimal de ce langage.

Examen

UE : Méthodes d'analyse et de conception des SI

Durée de l'épreuve 2h

Aucun document ni support autorisé.

1 Description du besoin

Promulguée le 12 juillet 2010, la loi portant engagement national pour l'environnement, dite « Grenelle 2 », est un texte d'application et de territorialisation du Grenelle Environnement et de la loi Grenelle 1. Un des chantiers de cette loi porte sur le développement des véhicules électriques et hybrides rechargeables, en favorisant l'émergence de l'offre industrielle nationale, en stimulant la demande et en encourageant la possibilité de créer et d'entretenir des infrastructures de recharge électrique nécessaires à l'usage de ces véhicules.

Bien que la plupart des infrastructures de recharge relève de la sphère privée (90%), les bornes de recharge accessibles au public, placées dans des parkings ou sur voirie, offriront l'assurance aux utilisateurs de pouvoir y accéder en dehors de cette sphère privée (domicile, travail) et des stations services. Elles constituent un gage de fiabilité de l'ensemble du système, complément indispensable pour encourager l'utilisation du véhicule électrique.

Les communes sont naturellement impliquées dans le déploiement de ces bornes, en raison du fort impact sur la voirie et les places de stationnement.

La ville de R. fait partie des douze agglomérations pilotes appelées à déployer une première vague d'infrastructures de recharge pour véhicules hybrides et électriques.

Elle a confié à la société prestataire de services Chargéon la mise en place et l'exploitation d'un réseau de points de recharge sous forme de bornes intelligentes standardisées. Soucieuse de se forger une expérience solide sur ce marché émergent, la société Chargéon a besoin de développer une application permettant de réaliser les principales opérations nécessaires au bon fonctionnement des différents sous-systèmes de l'infrastructure de recharge.

En tant qu'analyste-programmeur, vous participez au développement de cette application appelée CRAB – Chargement Rapide Automatisé de Batteries.

L'accès aux bornes de recharge par un usager est sujet à la souscription d'un contrat de recharge. Un portail *web* accessible aux usagers doit être mis en place pour faciliter le suivi de leur consommation.

Les bornes de recharge sont implantées dans des stations situées dans les parkings et sur la voirie de la ville. Chaque station accueille une ou plusieurs bornes et est localisée par ses coordonnées GPS (latitude et longitude) et l'adresse de la rue dans laquelle elle est située. Sont également mémorisées la date de mise en service de chaque borne et sa dernière date de révision.

Ces bornes de recharge de batteries sont préférentiellement du type « recharge normale » (puissance de 3 kW ou kilowatt) ou « semi-rapide » (puissance de 24 kW), mais certaines sont de type « recharge rapide » (puissance de 50 kW).

Afin de pouvoir utiliser les bornes de recharge mises à disposition par la mairie de R., tout possesseur d'un véhicule électrique doit souscrire un contrat de recharge référencant ce véhicule auprès des services municipaux. Les informations à renseigner sont les suivantes : nom, prénom, téléphones fixe et mobile, adresse postale, adresse de courriel, numéro d'immatriculation de la voiture, date du contrat. Il est également important de connaître le modèle de la batterie du véhicule (référence, capacité et fabricant). En effet, si toutes les batteries supportent le type de charge « normal », certaines ne supportent pas la charge « rapide », voire « semi-rapide ». Un usager peut bien entendu posséder plusieurs contrats. Pour chaque contrat, une carte magnétique est délivrée, cette carte permettra de se connecter aux bornes de recharge.

Les bornes de recharge comptabilisent l'énergie délivrée en kilowatt-heure (KWh).

Deux formules de contrat ont été retenues :

- Le forfait prépayé : l'utilisateur règle à l'avance un certain nombre de KWh. Les opérations de rechargement de batterie sont autorisées tant que le solde restant de KWh n'est pas épuisé. L'utilisateur peut à tout moment racheter des KWh.
- L'abonnement : l'utilisateur choisit une durée d'abonnement qui lui permettra de réaliser au cours de cette période autant d'opérations de rechargement que nécessaire. On mémorise les dates de début et de fin d'abonnement. L'utilisateur peut renouveler son abonnement, ce qui a pour effet de repousser la date de fin.

Après avoir immobilisé son véhicule, l'utilisateur présente sa carte magnétique auprès du lecteur de carte de la borne. Cette lecture permet alors d'identifier le contrat de recharge et son propriétaire. L'utilisateur déroule le câble de son chargeur de batterie et le connecte à la prise de la borne prévue à cet effet. Le système détermine alors le nombre de KWh nécessaires et informe l'utilisateur du temps de rechargement.

Plusieurs contrôles sont alors effectués :

- la borne détecte automatiquement les caractéristiques de la batterie du véhicule, qui doit correspondre au modèle déclaré dans le contrat ;
- la borne vérifie que la batterie connectée supporte bien le type de charge qu'elle délivre (« normal », « semi-rapide » ou « rapide ») ;
- la borne s'assure de la validité des dates du contrat dans le cas d'un abonnement ou du solde de KWh restant dans le cas de la formule prépayée.

L'échec d'un de ces contrôles bloque le rechargement ; cet échec est alors enregistré en regard de l'opération de rechargement en cours. À des fins statistiques, la cause de l'échec doit être mémorisée.

Si tous les contrôles sont positifs, le rechargement peut commencer.

À l'issue de l'opération, l'heure de début et de fin du rechargement ainsi que le nombre de KWh délivrés sont enregistrés.

Afin d'assurer la qualité de service attendues par les usagers, il s'agit d'optimiser la gestion des pannes pouvant survenir dans les infrastructures de recharge nécessaires à l'usage des véhicules électriques.

Toutes les interventions de maintenance corrective se font à la suite d'un incident déclaré.

Pour faciliter la planification des interventions, les types d'incidents les plus courants sont répertoriés dans la base de données avec une indication du temps de réparation prévu pour ce genre de panne. Une fois l'intervention réalisée par le technicien, l'incident est clôturé.

Il est nécessaire de conserver un historique des opérations dans une base accessible par tous les intervenants de maintenance. Cette base permet d'effectuer différentes statistiques aussi bien sur la fiabilité des bornes que sur la réactivité de la maintenance, dont l'analyse permettra d'augmenter la performance du système.

Les techniciens doivent également mener des actions de maintenance préventive sur les bornes de recharge. Ces révisions sont fonction du type de borne. Elles sont programmées à intervalles de temps réguliers, mais peuvent aussi être déclenchées lors de l'atteinte d'un seuil d'utilisation.

La solution informatique doit permettre chaque mois de répartir équitablement les tâches de maintenance préventive (visite) sur l'ensemble des techniciens.

Une visite concerne une station, et précise la durée totale nécessaire pour réaliser les révisions sur les bornes.

La répartition des visites aux différents techniciens doit être équitable en affectant chaque nouvelle visite au technicien actuellement le moins occupé en temps total de maintenance préventive.

2 Question

- 2.1 Décrivez les différentes étapes qui permettront d'aboutir au développement de cette nouvelle application. Précisez quelles tâches devront être réalisées à chaque étape et évaluez le temps que prendra chaque étape/sous-étape.**
- 2.2 A partir de l'énoncé ci-dessus, regroupez les fonctionnalités à réaliser en sous-systèmes**
- 2.3 Toujours à partir de l'énoncé ci-dessus, proposez le dictionnaire des données et un schéma conceptuel de la BD**

Module Framework – session 2 - année 2018-19
Yoann DIEUDONNE - Didier FERMENT - Université de Picardie

Durée : 2h00

Seuls documents autorisés : les polycopiés du cours, éventuellement annotés de votre main

des candidats aux élections européennes

Ce que j'ai déjà fait :

- création d'un projet, d'un bundle :

```
$ symfony new electionEuropeenne2 3.4
$ php bin/console generate:bundle
Bundle name: Election2Bundle
...
```

- création d'entités associées à une base de données :

```
<?php
namespace Election2Bundle\Entity;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Validator\Constraints as Assert;

/** @ORM\Table(name="candidat")
    @ORM\Entity(repositoryClass="Election2Bundle\Repository\CandidatRepository") */
class Candidat {
    /** @ORM\Column(name="id", type="integer") @ORM\Id
        @ORM\GeneratedValue(strategy="AUTO") */
    private $id;

    /** @ORM\Column(name="nom", type="string", length=30, unique=true) */
    private $nom;

    /** @ORM\Column(name="sexe", type="string", length=1) */
    private $sexe;

    /** @ORM\Column(name="age", type="smallint") */
    private $age;

    /** @ORM\ManyToOne(targetEntity="Election2Bundle\Entity\Liste",
        inversedBy="candidats", cascade={"persist"})
        @ORM\JoinColumn(onDelete="CASCADE") */
    private $liste;
}
```

et

```
<?php
namespace Election2Bundle\Entity;
use Doctrine\ORM\Mapping as ORM;

/** @ORM\Table(name="liste")
    @ORM\Entity(repositoryClass="Election2Bundle\Repository\ListeRepository") */
```

```

class Liste {
    /** @ORM\Column(name="id", type="integer") @ORM\Id
        @ORM\GeneratedValue(strategy="AUTO") */
    private $id;

    /** @ORM\Column(name="nom", type="string", length=30, unique=true) */
    private $nom;

    /** @ORM\OneToMany(targetEntity="Election2Bundle\Entity\Candidat",
        mappedBy="liste", cascade={"persist"}) */
    private $candidats;

    public function __toString() {
        return $this->nom;
    }
}

```

- puis génération complète des entités et mise à jour de la base de données :

```

$ php bin/console doctrine:generate:entities Election2Bundle
$ php bin/console doctrine:schema:update --force

```

- création des routes :

- app/config/routing.yml :

```

election2:
    resource: "@Election2Bundle/Resources/config/routing.yml"
    prefix: /election

```

- src/Election2Bundle/Resources/config/routing.yml :

```

election2_liste:
    resource: "@Election2Bundle/Resources/config/routing/liste.yml"
    prefix: /liste
election2_candidat:
    resource: "@Election2Bundle/Resources/config/routing/candidat.yml"
    prefix: /candidat
election2_question1:
    path: /test1
    defaults: { _controller: Election2Bundle:Default:test1 }

```

- src/Election2Bundle/Resources/config/routing/candidat.yml :

```

candidat_index:
    path: /
    defaults: { _controller: "Election2Bundle:Candidat:index" }
candidat_show:
    path: /{id}/show
    defaults: { _controller: "Election2Bundle:Candidat:show" }
candidat_new:
    path: /new
    defaults: { _controller: "Election2Bundle:Candidat:new" }
candidat_edit:
    path: /{id}/edit
    defaults: { _controller: "Election2Bundle:Candidat:edit" }
candidat_delete:

```

```

    path:    /{id}/delete
    defaults: { _controller: "Election2Bundle:Candidat:delete" }
candidat_feminine:
    path:    /feminine
    defaults: { _controller: "Election2Bundle:Candidat:feminine" }

```

- `src/Election2Bundle/Resources/config/routing/liste.yml` :

```

liste_index:
    path:    /
    defaults: { _controller: "Election2Bundle:Liste:index" }
liste_show:
    path:    /{id}/show
    defaults: { _controller: "Election2Bundle:Liste:show" }
liste_new:
    path:    /new
    defaults: { _controller: "Election2Bundle:Liste:new" }
liste_edit:
    path:    /{id}/edit
    defaults: { _controller: "Election2Bundle:Liste:edit" }
liste_delete:
    path:    /{id}/delete
    defaults: { _controller: "Election2Bundle:Liste:delete" }

```

- ajout du contrôleur `Election2Bundle/Controller/CandidatController.php` :

```

<?php
namespace Election2Bundle\Controller;

use Election2Bundle\Entity\Candidat;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Request;

class CandidatController extends Controller {

    public function indexAction() {
        $em = $this->getDoctrine()->getManager();
        $candidats = $em->getRepository('Election2Bundle:Candidat')->findAll();
        return $this->render('@Election2/Candidat/index.html.twig',
            array('candidats' => $candidats));
    }

    public function newAction(Request $request) {
        $candidat = new Candidat();
        $form = $this->createForm('Election2Bundle\Form\CandidatType', $candidat);
        $form->handleRequest($request);
        if ($form->isSubmitted() && $form->isValid()) {
            $em = $this->getDoctrine()->getManager();
            $em->persist($candidat);
            $em->flush();
            return $this->redirectToRoute('candidat_show',
                array('id' => $candidat->getId()));
        }
        return $this->render('@Election2/Candidat/new.html.twig',

```

```

        array('candidat' => $candidat, 'form' => $form->createView()));
    }

    public function showAction($id) {
        $candidat = $this->getCandidat($id);
        return $this->render('@Election2/Candidat/show.html.twig',
            array('candidat' => $candidat));
    }

    public function editAction(Request $request, $id) {
        $candidat = $this->getCandidat($id);
        $editForm = $this->createForm('Election2Bundle\Form\CandidatType', $candidat);
        $editForm->handleRequest($request);
        if ($editForm->isSubmitted() && $editForm->isValid()) {
            $this->getDoctrine()->getManager()->flush();
            return $this->redirectToRoute('candidat_edit',
                array('id' => $candidat->getId()));
        }
        return $this->render('@Election2/Candidat/edit.html.twig',
            array('candidat' => $candidat,
                'edit_form' => $editForm->createView()));
    }

    public function deleteAction(Request $request, $id) {
        $candidat = $this->getCandidat($id);
        $em = $this->getDoctrine()->getManager();
        $em->remove($candidat);
        $em->flush();
        return $this->redirectToRoute('candidat_index');
    }

    private function getCandidat($id) { // retourne l'objet candidat à partir de son id
        $repository = $this->getDoctrine()->getManager()
            ->getRepository('Election2Bundle:Candidat');
        $candidat = $repository->find($id);
        if ($candidat == null)
            throw $this->createNotFoundException('candidat inexistant');
        return $candidat;
    }
}

```

• ses vues :

- Election2Bundle/Resources/views/Candidat/index.html.twig :

```

{% extends 'base.html.twig' %}
{% block body %}
    <h1>Candidats list</h1>
    <ul> {% for candidat in candidats %}
        <li>{{ candidat.nom }} sexe:{{ candidat.sexe }} age={{ candidat.age }}
            <a href="{{ path('candidat_show', { 'id': candidat.id }) }}">show</a></li>
        {% endfor %}
    </ul>

```

```
<p><a href="{{ path('candidat_new') }}">Create a new candidat</a></p>
{% endblock %}
```

◦ Election2Bundle/Resources/views/Candidat/show.html.twig :

```
{% extends 'base.html.twig' %}
```

```
{% block body %}
```

```
<h1>Candidat</h1>
```

```
<ul>
```

```
<li>Nom : {{ candidat.nom }}</li>
```

```
<li>Sexe : {{ candidat.sexe }}</li>
```

```
<li>Age = {{ candidat.age }}</li>
```

```
<li><a href="{{ path('candidat_index') }}">Back to the list</a></li>
```

```
<li><a href="{{ path('candidat_edit', { 'id': candidat.id }) }}">Edit</a></li>
```

```
</ul>
```

```
{% endblock %}
```

◦ Election2Bundle/Resources/views/Candidat/edit.html.twig :

```
{% extends 'base.html.twig' %}
```

```
{% block body %}
```

```
<h1>Candidat edit</h1>
```

```
{{ form_start(edit_form) }}
```

```
    {{ form_widget(edit_form) }}
```

```
    <input type="submit" value="Edit" />
```

```
{{ form_end(edit_form) }}
```

```
<ul>
```

```
<li><a href="{{ path('candidat_index') }}">Back to the list</a></li>
```

```
<li><a href="{{ path('candidat_delete',
                                { 'id': candidat.id }) }}">delete</a></li>
```

```
</ul>
```

```
{% endblock %}
```

◦ Election2Bundle/Resources/views/Candidat/new.html.twig :

```
{% extends 'base.html.twig' %}
```

```
{% block body %}
```

```
<h1>Candidat creation</h1>
```

```
{{ form_start(form) }}
```

```
    {{ form_widget(form) }}
```

```
    <input type="submit" value="Create" />
```

```
{{ form_end(form) }}
```

```
<p><a href="{{ path('candidat_index') }}">Back to the list</a></p>
```

```
{% endblock %}
```

• le "form type" Election2Bundle/Form/Candidat_Type.php :

```
<?php
```

```
namespace Election2Bundle\Form;
```

```
use Symfony\Component\Form\AbstractType;
```

```
use Symfony\Component\Form\FormBuilderInterface;
```

```
use Symfony\Component\OptionsResolver\OptionsResolver;
```

```
class CandidatType extends AbstractType {
```

```
    public function buildForm(FormBuilderInterface $builder, array $options) {
```

```
        $builder->add('nom')->add('sexe')->add('age')->add('liste');
```

```
    }
```

```
    public function configureOptions(OptionsResolver $resolver) {
```

```
        $resolver->setDefaults(array('data_class'=>
```

```
'Election2Bundle\Entity\Candidat'));
```

```
}
```

```
}
```

- ajout du contrôleur Election2Bundle/Controller/ListeController.php :

```
<?php
```

```
namespace Election2Bundle\Controller;
```

```
use Election2Bundle\Entity>Liste;
```

```
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
```

```
use Symfony\Component\HttpFoundation\Request;
```

```
class ListeController extends Controller {
```

```
    public function indexAction() {
```

```
        $em = $this->getDoctrine()->getManager();
```

```
        $listes = $em->getRepository('Election2Bundle:Liste')->findAll();
```

```
        return $this->render('@Election2/Liste/index.html.twig', array(
            'listes' => $listes));
```

```
    }
```

```
    public function newAction(Request $request) {
```

```
        $liste = new Liste();
```

```
        $form = $this->createForm('Election2Bundle\Form>ListeType', $liste);
```

```
        $form->handleRequest($request);
```

```
        if ($form->isSubmitted() && $form->isValid()) {
```

```
            $em = $this->getDoctrine()->getManager();
```

```
            $em->persist($liste);
```

```
            $em->flush();
```

```
            return $this->redirectToRoute('liste_show',
                array('id' => $liste->getId()));
```

```
        }
```

```
        return $this->render('@Election2/Liste/new.html.twig',
```

```
            array('liste' => $liste, 'form' => $form->createView()));
```

```
    }
```

```
    public function showAction($id) {
```

```
        $liste = $this->getListe($id);
```

```
        return $this->render('@Election2/Liste/show.html.twig',
            array('liste' => $liste));
```

```
    }
```

```
    public function editAction(Request $request, $id) {
```

```
        $liste = $this->getListe($id);
```

```
        $editForm = $this->createForm('Election2Bundle\Form>ListeType', $liste);
```

```
        $editForm->handleRequest($request);
```

```
        if ($editForm->isSubmitted() && $editForm->isValid()) {
```

```
            $this->getDoctrine()->getManager()->flush();
```

```
            return $this->redirectToRoute('liste_edit', array('id' => $id));
```

```
        }
```

```
        return $this->render('@Election2/Liste/edit.html.twig',
```

```
            array('liste' => $liste, 'edit_form' => $editForm->createView()));
```

```
    }
```

```

public function deleteAction(Request $request, $id) {
    $liste = $this->getListe($id);
    $em = $this->getDoctrine()->getManager();
    $em->remove($liste);
    $em->flush();
    return $this->redirectToRoute('liste_index');
}

private function getListe($id) { // retourne l'entité liste à partir de son id
    $repository = $this->getDoctrine()->getManager()
        ->getRepository('Election2Bundle:Liste');
    $liste = $repository->find($id);
    if ($liste == null)
        throw $this->createNotFoundException('liste inexistant');
    return $liste;
}

public function __toString() {
    return $this->getNom();
}
}

```

• ses vues :

◦ Election2Bundle/Resources/views/Liste/index.html.twig :

```

{% extends 'base.html.twig' %}
{% block body %}
    <h1>Listes list</h1>
    <ul> {% for liste in listes %}
        <li>{{ liste.nom }}
            <a href="{{ path('liste_show', { 'id': liste.id }) }}">show</a></li>
        {% endfor %}
    </ul>
    <p><a href="{{ path('liste_new') }}">Create a new liste</a></p>
{% endblock %}

```

◦ Election2Bundle/Resources/views/Liste/show.html.twig :

```

{% extends 'base.html.twig' %}
{% block body %}
    <h1>Liste</h1>
    <ul>
        <li>Nom : {{ liste.nom }}</li>
        <li>candidats :
            <ul>
                {% for candidat in liste.candidats %}
                    <li>{{ candidat.nom }}</li>
                {% endfor %}
            </ul>
        </li>
        <li><a href="{{ path('liste_index') }}">Back to the list</a></li>
        <li><a href="{{ path('liste_edit', { 'id': liste.id }) }}">Edit</a></li>
    </ul>

```

```
{% endblock %}
```

- Election2Bundle/Resources/views/Liste/edit.html.twig :

```
{% extends 'base.html.twig' %}
```

```
{% block body %}
```

```
<h1>Liste edit</h1>
```

```
{{ form_start(edit_form) }}
```

```
    {{ form_widget(edit_form) }}
```

```
    <input type="submit" value="Edit" />
```

```
{{ form_end(edit_form) }}
```

```
<ul>
```

```
    <li><a href="{{ path('liste_index') }}">Back to the list</a></li>
```

```
    <li><a href="{{ path('liste_delete', { 'id': liste.id }) }}">delete</a></li>
```

```
</ul>
```

```
{% endblock %}
```

- Election2Bundle/Resources/views/Liste/new.html.twig :

```
{% extends 'base.html.twig' %}
```

```
{% block body %}
```

```
<h1>Liste creation</h1>
```

```
{{ form_start(form) }}
```

```
    {{ form_widget(form) }}
```

```
    <input type="submit" value="Create" />
```

```
{{ form_end(form) }}
```

```
<p><a href="{{ path('liste_index') }}">Back to the list</a></p>
```

```
{% endblock %}
```

- le "form type" Election2Bundle/Form/Liste_Type.php :

```
<?php
```

```
namespace Election2Bundle\Form;
```

```
use Symfony\Component\Form\AbstractType;
```

```
use Symfony\Component\Form\FormBuilderInterface;
```

```
use Symfony\Component\OptionsResolver\OptionsResolver;
```

```
class ListeType extends AbstractType {
```

```
    public function buildForm(FormBuilderInterface $builder, array $options) {
```

```
        $builder->add('nom');
```

```
    }
```

```
    public function configureOptions(OptionsResolver $resolver) {
```

```
        $resolver->setDefaults(array(
```

```
            'data_class' => 'Election2Bundle\Entity\Liste');
```

```
        )
```

```
    }
```

Question 1 :

J'ajoute le code ci-dessous au DefaultController.php :

```
<?php
namespace Election2Bundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Election2Bundle\Entity\Liste;
use Election2Bundle\Entity\Candidat;

class DefaultController extends Controller {
    public function test1Action() {
        $em = $this->getDoctrine()->getManager();
        $candidat1 = new Candidat();
        $candidat1->setNom("Dupont Saignant");
        $candidat1->setAge(62);
        $candidat1->setSexe("M");
        $em->persist($candidat1);
        $candidat2 = new Candidat();
        $candidat2->setNom("Inelligible");
        $candidat2->setAge(29);
        $candidat2->setSexe("F");
        $em->persist($candidat2);
        $liste = new Liste();
        $liste->setNom("Debout en Transe");
        $candidat1->setListe($liste);
        $liste->addCandidat($candidat2);
        $em->flush();
        return $this->redirectToRoute('liste_show', array('id' => $liste->getId()));
    }
}
```

Qu'est-ce qui est affiché par le navigateur pour l'URL <http://localhost:8000/election/test1> ?

Question 2 :

La suppression d'une liste via la route "liste_delete" fonctionne t'elle normalement ?

Provoque t'elle une erreur ?

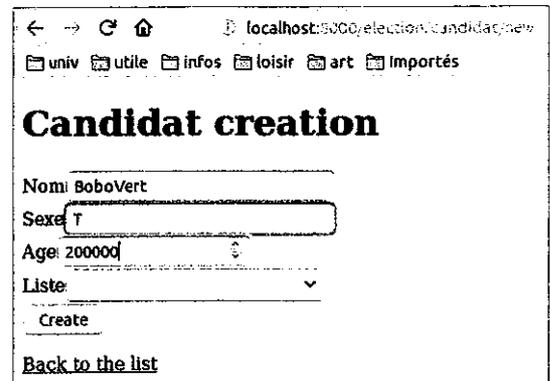
Si non, pourquoi ?

Si oui, quelle correction doit être apportée au code ?

Question 3 :

Comment éviter que des rigolos comme ce "BoboVert" de sexe "T(rans)" et d'age délirant puisse être ajouté dans notre base de données ?

Le sexe est F ou M exclusivement, et l'age sur 2 à 3 chiffres avec un minimum de 18.



localhost:5000/election/candidat/new

univ utile infos loisir art Importés

Candidat creation

Nom: Bobovert

Sexe: T

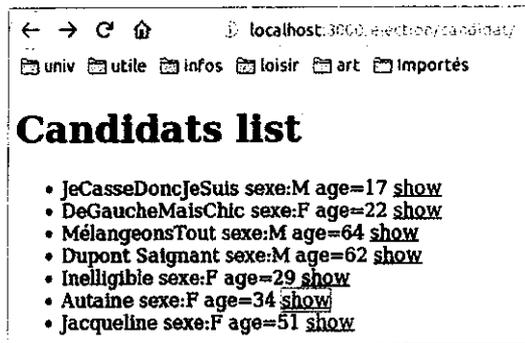
Age: 200000

Liste: ▾

Create

[Back to the list](#)

Question 4 :



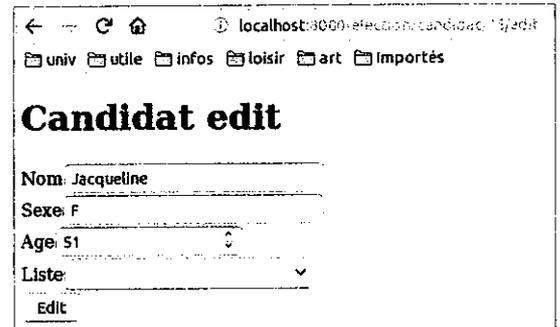
localhost:3000/election/candidat/

univ utile infos loisir art Importés

Candidats list

- JeCasseDoncJeSuis sexe:M age=17 [show](#)
- DeGaucheMaisChic sexe:F age=22 [show](#)
- MélangeonsTout sexe:M age=64 [show](#)
- Dupont Saignant sexe:M age=62 [show](#)
- Inelligible sexe:F age=29 [show](#)
- Autaine sexe:F age=34 [show](#)
- Jacqueline sexe:F age=51 [show](#)

Voici la liste des candidats actuels, dont "Jacqueline" qui ne figure dans aucune liste



localhost:3000/election/candidat/1/edit

univ utile infos loisir art Importés

Candidat edit

Nom: Jacqueline

Sexe: F

Age: 51

Liste:

Edit

Écrire le(s) code(s) de l'action "feminine" qui affiche toutes les candidates figurant dans une liste à l'élection : nom, age, et liste (avec un lien sur 'liste_show'). Ci-dessous un exemple :
Le routing est déjà écrit.



localhost:3000/election/candidat/feminine

univ utile infos loisir art Importés

Candidates figurant dans une liste pour l'élection

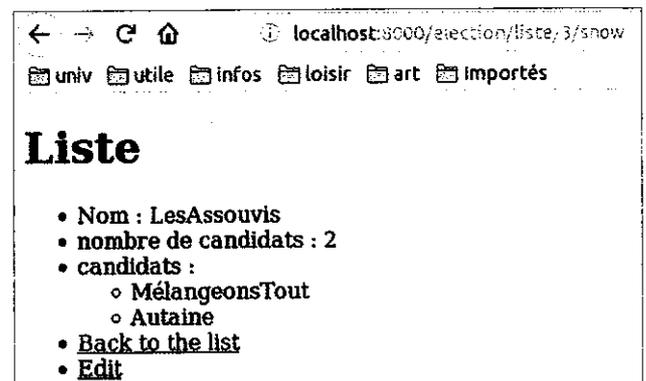
- DeGaucheMaisChic age=22 [LaRépubliqueAMarcheForcée](#)
- Autaine age=34 [LesAssouvis](#)

Question 5 :

Je voudrais modifier l'affichage (liste_show) de la liste pour lui ajouter son nombre de candidats, comme ci-contre :

Je modifie ainsi sa vue :

```
{% extends 'base.html.twig' %}
{% block body %}
  <h1>Liste</h1>
  <ul>
    <li>Nom : {{ liste.nom }}</li>
    <li>nombre de candidats : {{ nombre }} </li>
    <li>candidates :
      <ul>
        {% for candidat in liste.candidates %}
          <li>{{ candidat.nom }}</li>
        {% endfor %}
      </ul>
    </li>
    <li><a href="{{ path('liste_index') }}">Back to the list</a></li>
    <li><a href="{{ path('liste_edit', { 'id': liste.id }) }}">Edit</a></li>
  </ul>
{% endblock %}
```



localhost:3000/election/liste/3/show

univ utile infos loisir art Importés

Liste

- Nom : LesAssouvis
- nombre de candidats : 2
- candidats :
 - MélangeonsTout
 - Autaine
- [Back to the list](#)
- [Edit](#)

et l'action ainsi :

```
public function showAction($id) {
    $liste = $this->getListe($id);
```

```
$repo = $this->getDoctrine()->getManager()  
        ->getRepository('Election2Bundle:Candidat');  
$nombre = $repo->getNombreCandidats($id);  
return $this->render('@Election2/Liste/show.html.twig',  
        array('liste' => $liste, 'nombre' => $nombre));  
}
```

Que faut-il encore écrire comme code ? Donnez-le.

Session de Juin 2019
Méthodes Quantitatives et Décision(*) – Durée 2h

Exercice 1

Un particulier souhaite préparer un gâteau à base de trois fruits. Son état de santé l'oblige à respecter certaines contraintes liées aux doses de glucides, de vitamines et de magnésium. Le dosage, de 100 g de chacun des fruits utilisés, est représenté dans le tableau suivant :

| Fruit | Glucides (g) | Vitamine (mg) | Magnésium (mg) |
|--------|--------------|---------------|----------------|
| Papaye | 11.3 | 4 | 0.9 |
| Kiwi | 9 | 15 | 0 |
| Orange | 18.3 | 30 | 0.7 |

Son médecin traitant lui recommande, pour une portion de 100 g, un maximum de 10 g de glucides, au moins 15 mg de vitamines et un minimum de 0.8 mg de magnésium.

Notons que le gâteau à préparer pèse un kilogramme et le prix d'achat, pour 100 g de fruits, est de 4.5 euros pour le premier fruit, de 3.8 euros pour le deuxième fruit et de 2.6 euros pour le troisième fruit (selon l'ordre indiqué dans le tableau ci-dessus).

- a) Formuler le programme linéaire permettant de déterminer les fruits à acheter à moindre coût, tout en respectant les recommandations du médecin traitant.
- b) Pour le programme de la question (a), proposer une solution admissible de départ afin d'appliquer la méthode du simplexe.
- c) En appliquant la méthode du simplexe, montrer que le problème des questions (a) et (b) admettent au moins une solution optimale. Dans ce cas, interpréter le régime associé à cette solution.
- d) Est-il possible de résoudre graphiquement le programme précédemment formulé?

Exercice 2

Lors de la résolution d'un programme linéaire (de maximisation) avec l'algorithme du Simplexe, nous obtenons le tableau suivant :

| x_1 | x_2 | x_3 | x_4 | β |
|-------|-------|--------------|--------------|-------------------------|
| 0 | 1 | 0 | 1 | $3 + \frac{1}{2}\alpha$ |
| 1 | 0 | -1 | 1 | 1 |
| 0 | 0 | $\alpha + 2$ | $\alpha + 1$ | 2 |

Question : Discuter en fonction du paramètre α , la solution courante du tableau associé à un programme linéaire.

*Cours autorisé (Barème indicatif : exo 1 (6-7 pts), exo 2 (4 pts), exo 3 (4-5 pts) et exo 4 (6-7 pts)).

Exercice 3

Considérons le problème suivant :

$$\left\{ \begin{array}{l} \text{Min } z = |x_3 - 5x_1| + 7x_2 \\ \text{s.c. } 4x_1 - x_2 + 2x_3 = 6 \\ \phantom{\text{s.c.}} 2x_2 - 4x_3 \geq 4 \\ \phantom{\text{s.c.}} x_1, x_3 \geq 0 \end{array} \right.$$

- a) Linéariser ce problème.
- b) Donner la forme canonique puis la forme standard du programme linéaire de la question (a).

Exercice 4

Soit le programme linéaire suivant :

$$\left\{ \begin{array}{l} \text{Min } z = -2x_1 + x_2 \\ \text{s.c. } x_2 \leq x_1 \leq x_2 + 2 \\ \phantom{\text{s.c.}} 2 \leq x_1 \leq 8 \end{array} \right.$$

1. Déterminer graphiquement l'espace des solutions admissibles de ce programme.
2. En introduisant les courbes de niveaux associées à la fonction objectif z , déterminer une solution optimale pour ce programme (utiliser le gradient pour l'objectif).
3. Par application de la méthode du simplexe, résoudre le programme linéaire.
4. Comparer les solutions (et évaluations) obtenues de (2) et (3).

Examen de XML

Tous les documents au format papier sont autorisés (cela inclut les notes de cours et TP).

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

Ci-dessous se trouve le fichier *catalog.xml*. Observez-le attentivement car il sera utilisé de manière récurrente dans les 3 exercices qui suivent. Veuillez noter qu'entre deux balises, il y a au moins un espace ou un retour chariot.

```
<?xml version="1.0"?>
<?xml-stylesheet type="application/xml" href="transform.xsl"?>
<catalog val="0">
  <book id="bk101" val="1">
    <author val="2">Gambardella, Matthew</author>
    <title val="3">XML Developer's Guide</title>
    <genre val="4">Computer</genre> <price val="5">44.95</price>
    <publish_date val="6">2000-10-01</publish_date>
    <description val="7">An in-depth look at creating applications
with XML.</description>
  </book>
  <book id="bk102" val="8">
    <author val="9">Ralls, Kim</author>
    <title val="10">Midnight Rain</title>
    <genre val="11">Fantasy</genre> <price val="12">6.95</price>
    <publish_date val="13">2000-12-16</publish_date>
    <description val="14">A former architect battles corporate zombies,
an evil sorceress, and her own childhood to become queen
of the world.</description>
  </book>
  <book id="bk103" val="15">
    <author val="16">Corets, Eva</author>
    <title val="17">Maeve Ascendant</title>
    <genre val="18">Fantasy</genre> <price val="19">5.75</price>
    <publish_date val="20">2000-11-17</publish_date>
    <description val="21">After the collapse of a nanotechnology
society in England, the young survivors lay the
foundation for a new society.</description>
  </book>
  <book id="bk104" val="22">
    <author val="23">Corets, Eva</author>
    <title val="24">Oberon's Legacy</title>
    <genre val="25">Fantasy</genre> <price val="26">3.95</price>
    <publish_date val="27">2001-03-10</publish_date>
    <description val="28">In post-apocalypse England, the mysterious
agent known only as Oberon helps to create a new life
for the inhabitants of London. Sequel to Maeve
Ascendant.</description>
  </book>
  <book id="bk105" val="29">
    <author val="30">Corets, Eva</author>
    <title val="31">The Sundered Grail</title>
    <genre val="32">Fantasy</genre> <price val="33">5.95</price>
    <publish_date val="34">2001-09-10</publish_date>
```

```
<description val="35">The two daughters of Maeve, half-sisters,
battle one another for control of England. Sequel to
Oberon's Legacy.</description>
</book>
<book id="bk106" val="36">
  <author val="37">Randall, Cynthia</author>
  <title val="38">Lover Birds</title>
  <genre val="39">Romance</genre> <price val="40">4.99</price>
  <publish_date val="41">2000-09-02</publish_date>
  <description val="42">When Carla meets Paul at an ornithology
conference, tempers fly as feathers get ruffled.</description>
</book>
<book id="bk107" val="43">
  <author val="44">Thurman, Paula</author>
  <title val="45">Splish Splash</title>
  <genre val="46">Romance</genre> <price val="47">4.95</price>
  <publish_date val="48">to_appear</publish_date>
  <description val="49">A deep sea diver finds true love twenty
thousand leagues beneath the sea.</description>
</book>
<book id="bk108" val="50">
  <author val="51">Knorr, Stefan</author>
  <title val="52">Creepy Crawlies</title>
  <genre val="53">Horror</genre> <price val="54">4.95</price>
  <publish_date val="55">2000-12-06</publish_date>
  <description val="56">An anthology of horror stories about roaches,
centipedes, scorpions and other insects.</description>
</book>
<book id="bk109" val="57">
  <author val="58">Kress, Peter</author>
  <title val="59">Paradox Lost</title>
  <genre val="60">Science Fiction</genre> <price val="61">6.90</price>
  <publish_date val="62">2000-11-02</publish_date>
  <description val="63">After an inadvertant trip through a Heisenberg
Uncertainty Device, James Salway discovers the problems
of being quantum.</description>
</book>
<book id="bk110" val="64">
  <author val="65">O'Brien, Tim</author>
  <title val="66">Microsoft .NET: The Programming Bible</title>
  <genre val="67">Computer</genre> <price val="68">38.95</price>
  <publish_date val="69">to_appear</publish_date>
  <description val="70">Microsoft's .NET initiative is explored in
detail in this deep programmer's reference.</description>
</book>
<book id="bk111" val="71">
  <author val="72">O'Brien, Tim</author>
  <title val="73">MSXML3: A Comprehensive Guide</title>
  <genre val="74">Computer</genre> <price val="75">36.95</price>
  <publish_date val="76">2000-12-01</publish_date>
  <description val="77">The Microsoft MSXML3 parser is covered in
detail, with attention to XML DOM interfaces, XSLT processing,
SAX and more.</description>
</book>
</catalog>
```

Exercice 1 (7 points) : Ci-dessous se trouve le code incomplet d'un schéma XML : il manque 2 parties (code1 et code2) indiquées en gras. Donnez les deux parties manquantes de façon à ce qu'au final on obtienne un schéma pour lequel le fichier *catalog.xml* est valide et pour lequel les conditions suivantes sont imposées :

- 1) Les attributs *id* des éléments *book* sont des identifiants.
- 2) L'élément *genre* ne peut contenir que un des textes de l'ensemble suivant {Computer; Fantasy; Romance; Horror; Science Fiction}.
- 3) Deux éléments *book* ne peuvent pas avoir le même titre et le même auteur.
- 4) L'élément *publish_date* doit être défini sans passer par une expression régulière : il doit contenir soit une date au format classique *dd-mm-yyyy* soit le mot clé *to_appear*.
- 5) Deux noeuds éléments quelconques ne peuvent pas avoir le même attribut *val*.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="StringWithVal">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="val" type="xsd:int" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:complexType name="PriceWithVal">
    <xsd:simpleContent>
      <xsd:restriction base="StringWithVal">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="\d+(\.\d{1,2})?" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:simpleType name="DateOrToAppear">
    <!-- code1 à compléter -->
  </xsd:simpleType>

  <xsd:complexType name="DateOrToAppearWithVal">
    <xsd:simpleContent>
      <xsd:extension base="DateOrToAppear">
        <xsd:attribute name="val" type="xsd:int" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
```

```
<xsd:element name="catalog">
  <!-- code2 à compléter -->
</xsd:element>
</xsd:schema>
```

Exercice 2 (6 points) :

- a) Donnez la valeur des attributs *val* des nœuds (du document *catalog.xml*) désignés par le chemin suivant : `//.././././.*`
- b) Donnez la valeur des attributs *val* des nœuds (du document *catalog.xml*) désignés par le chemin suivant : `//catalog/book[position() mod 3 =0]`
- c) Donnez une expression XPath permettant de sélectionner tous les titres des livres dont le prix est supérieur à 35.
- d) Donnez une expression XPath permettant de sélectionner les livres dont le prix est supérieur à 5 euros et les livres dont le genre est « Romance ».
- e) Donnez une expression XPath permettant d'indiquer le nombre de livres présents dans le document xml qui n'ont pas été écrits par « O'Brien, Tim ».
- f) Donnez une expression XPath permettant d'indiquer la somme des prix des livres écrits par « Corets, Eva »

Exercice 3 (7 points) : Donnez le code xml qui est produit à partir du fichier catalog.xml lorsque le code de transform.xml est celui donné plus bas.

Remarque : le nouveau document xml est normalement non indenté : cependant, afin d'améliorer la lisibilité mais aussi faciliter l'écriture, il est fortement conseillé de rajouter des retours à la ligne et/ou des espaces supplémentaires lors de la rédaction du document xml obtenu.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" />
  <xsl:template match="description">
    <xsl:element name="contenu"> <xsl:value-of select="description" />
  </xsl:element>
</xsl:template>
<xsl:template match="genre">
  <xsl:element name="genre"> <xsl:value-of select="genre" />
</xsl:element>
</xsl:template>
<xsl:template match="catalog">
  <xsl:element name="catalogue">
    <xsl:apply-templates select="//author[not(preceding::author=.)]" />
    <xsl:apply-templates select="*" />
  </xsl:element>
</xsl:template>
<xsl:template name="mystere">
  <xsl:param name="truc" />
  <xsl:for-each select="//descendant::author[.= $truc][1]">
    <xsl:variable name="nombre" select="count(preceding::author[not(preceding::author=.)])"/>
    <xsl:variable name="nombre2" select="$nombre+1"/>
    <xsl:value-of select="concat('a',$nombre2)" />
  </xsl:for-each>
</xsl:template>
<xsl:template match="author">
  <xsl:element name="auteur">
    <xsl:attribute name="pif"> <xsl:value-of select="concat('a',position())" />
  </xsl:attribute>
  <xsl:value-of select="child::text()"/>
</xsl:element>
</xsl:template>
<xsl:template match="book">
  <xsl:element name="livre">
    <xsl:attribute name="titre"> <xsl:value-of select="title" />
  </xsl:attribute>
  <xsl:attribute name="bidule">
    <xsl:call-template name="mystere">
      <xsl:with-param name="truc" select="author" />
    </xsl:call-template>
  </xsl:attribute>
</xsl:element>
</xsl:template>
</xsl:stylesheet>
```

PARTIEL RATTRAPAGE

MIAGE

(Durée : 2h)

DOSSIER N°1 :

L'entreprise GERVAIS vend un modèle unique de Trottinette électrique. Elle réalise un chiffre d'affaires annuel correspondant à 32000 trottinettes vendus à 110 € HT l'unité, ses charges fixes sont de 1.800.000 € et les charges variables représentent 40 % du chiffre d'affaires.

QUESTIONS

1. Présentez le compte de résultat différentiel
2. Calculez le seuil de rentabilité de l'entreprise Gervais d'après les données ci-dessus.
3. Combien doit-elle vendre de trottinettes pour atteindre son seuil de rentabilité ?
4. Déterminez la date du point mort.
5. Que pensez-vous de l'indice de sécurité de Gervais ?
6. L'entreprise Gervais souhaite avoir un résultat de 500 000 € l'année prochaine, quel doit être son niveau de chiffre d'affaires ?
7. En sachant que le coefficient d'élasticité est de -2,1, quel serait l'impact d'une baisse du prix des trottinette de 15% sur le chiffre d'affaires de l'entreprise ?
8. L'entreprise Gervais prévoit un investissement de 840 000 € (durée de vie 7 ans) afin d'assurer une augmentation de 15% de son chiffre d'affaires. Son seuil de rentabilité a-t-il changé ? Faire le calcul le cas échéant.

DOSSIER N°2 : MISE EN ŒUVRE DE LA METHODE DES COÛTS COMPLETS

La société D'ARTBOIS commercialise deux produits qu'elle fabrique : des tables en bois et des comptoirs bois/inox. L'ensemble des opérations de menuiseries, depuis le débit du bois brut jusqu'à l'assemblage des meubles, sont effectuées dans un même atelier. Les comptoirs bois reçoivent ensuite le revêtement d'une feuille inox préparé dans un second atelier.

A l'exception des matières consommables et des accessoires de quincaillerie, l'entreprise réalise un inventaire mensuel des matières premières et des fabrications de chaque atelier.

| DONNEES DE FABRICATION (quantités par unité produite) | TABLES | | COMPTOIRS | |
|--|--|--------------|---|--|
| | ATELIER BOIS | ATELIER INOX | ATELIER BOIS | ATELIER INOX |
| - consommation bois (hêtre) - consommation tôle inox 2mm - main d'œuvre directe (M.O.D.) - produits non stockés (consommables et quincaillerie) | 0,25 m ³ 3/4 d'heure à 22 €/h 4,47 € | | 0,85 m ³ 2 heures à 22 €/h 6,95 € | 0,25 kg 3/4 d'heure à 24 €/h 9,21 € |

| ETAT DES STOCKS (mois de JUIN) | STOCK INITIAL | | STOCK FINAL | |
|--|---------------|--------|-------------|-------|
| | QUANTITES | P. U. | TOTAL | TOTAL |
| - bois (m ³) | 800 | 179,17 | 143 336,00 | 0,00 |
| - inox (kg) | 1300 | 8,55 | 11 115,00 | 0,00 |
| - tables (produits finis) | 800 | 75,75 | 60 600,00 | 0,00 |
| - comptoirs bois (produits semi-finis) | 75 | 274,65 | 20 598,75 | 0,00 |
| - comptoirs (produits finis) | 160 | 335,00 | 53 600,00 | 0,00 |

| ACHATS MATIERES PREMIERES (mois de JUIN) | QUANTITES | P. U. |
|---|--------------------|--------|
| - hêtre brut de scierie (m ³) | 600 m ³ | 178,00 |
| - tôle inox épaisseur 2mm (kg) | 8 000 kg | 7,10 |

| PRODUCTION ATELIERS (mois de JUIN) | ATELIER BOIS | ATELIER INOX |
|---------------------------------------|--------------|--------------|
| nombre d'unités produites | | |
| - nombre de tables | 1460 unités | |
| - nombre de comptoirs | 220 unités | 240 unités |

| VENTES (mois de JUIN) | QUANTITES | P. U. |
|---|-------------|---------------------|
| produits finis : | | |
| - tables bois (u) | 1830 unités | 90 € |
| - comptoirs bois/inox (u) | 270 unités | 370 € |
| vente de déchets industriels : | | |
| 10% du bois consommé en atelier (m ³) | | 16 €/m ³ |
| 5% de inox consommé en atelier (kg) | | 3,60 €/kg |

TRAVAIL A EFFECTUER :

- 1 : établir le schéma de production de l'activité de l'entreprise d'ARTBOIS
- 2 : établir les tableaux d'analyse des coûts et de compte des stocks afin de déterminer le résultat analytique de l'entreprise au mois de juin selon la méthode des d'analyse des coûts complets
- 3 : renseigner dans l'état de stock la valorisation des stocks fin juin

nota : les calculs de valeurs monétaires seront arrondis à la seconde décimale

Université de Picardie Jules Verne

UFR des Sciences

Module Réseau

L3 Info & Miage

Examen de Réseaux: 27 juin 2019

Durée: 2 heures

(Documents non autorisés)

(L'usage de la calculatrice est interdit)

(Les téléphones portables doivent être éteints)

Exercice 1.

1. Quelle est la rapidité de modulation nécessaire pour que le canal de transmission ait un débit binaire de 2400 bits/s, sachant que les signaux transmis sont binaires? (1pt)
2. Pour obtenir ce même débit, quelle devrait être la valeur minimum du rapport signal/bruit si la largeur de la bande passante de la liaison est de 1000 Hz. (1pt)
3. Dans un réseau, un nœud N a 3 lignes en entrée avec les débits D_1 , D_2 et D_3 et une seule ligne à la sortie avec un débit D_s . Le buffer d'entrée est de taille bornée. Donnez la condition pour que le nœud N ne soit pas congestionné. (1pt)

Exercice 2

On considère un code linéaire $C(6, 3)$ dont la matrice génératrice de code est

$$G = \begin{bmatrix} 100011 \\ 010101 \\ 001110 \end{bmatrix}$$

1. $X_1 = 010101$ et $X_2 = 111101$ sont-ils des mots de code? Justifier clairement votre réponse. (On ne vous demande pas de déterminer les mots de code dans cette question; toute tentative dans ce sens vaudra une note zéro pour la question) (2pts)
2. Déterminer tous les mots de code. (1pts)
3. Quelle est la distance de code? (1 pt)
4. Combien d'erreurs peut-il corriger? (1 pt)

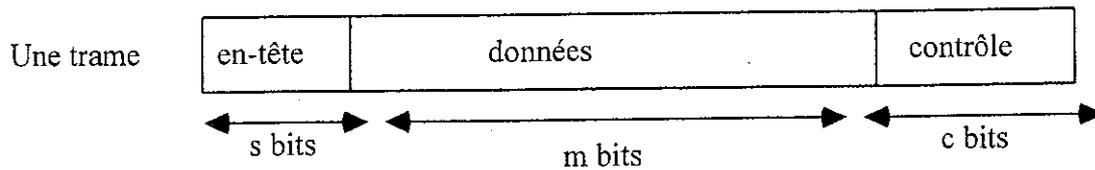
Exercice 3

On utilise le polynôme $g(x) = x^4 + x + 1$ pour coder des données constituées de 3 bits. Le codage est fait sous forme systématique.

1. Donner les mots de code associés aux données suivantes: $U_1 = 110$, $U_2 = 100001$ et $U_3 = 11111$. (1 pt)
2. On a reçu successivement 0110101 et 0111100. Quelles conclusions concernant les données émises et la transmission peut-on en tirer? (1 pt)

Exercice 4 (5pts)

Stop and Wait Liaison half duplex

Soit D le débit brut de la ligne τ : taux d'erreurs t_p : temps de propagation entre les deux stations t_r : temps de retournement du modem1 trame d'acquittement a q bits1) Quelle est la durée du cycle θ entre 2 émissions?2) Les erreurs étant indépendantes, quelle est la probabilité P_c que la trame et l'acquittement arrivent correctement?3) Calculer θ : durée moyenne pour transmettre une trame de façon correcte.4) a. Calculer le débit utile D_u (ou débit efficace) moyen i.e. le rapport entre le nombre de bits d'information transmis sur le temps moyen mis.b. Trouver la valeur de m qui maximalise D_u .**Exercice 5** (5pts)

On considère un support de transmission dont l'accès par les stations utilise le protocole *AMRT ou TDMA dynamique*.

Exécuter ce protocole sur l'exemple suivant (en indiquant sur chaque slot le propriétaire et l'utilisateur, et s'il y a eu un tirage aléatoire pour prendre ce slot). Les tirages de la station 1 sont NON, NON, NON, OUI, OUI, NON, OUI, OUI et les tirages de la station 3 sont NON, OUI, NON, NON. On note :

- $_t(i)$ quand il y a eu un tirage aléatoire défavorable de la station i sur un slot
- $_i_t(i)$ quand il y a eu un tirage aléatoire favorable de la station i sur un slot
- $_t(i, j)$ quand il y a eu des tirages aléatoires défavorables des station i et j sur le même slot
- $_i_t(i, j)$ quand il y a eu deux tirages aléatoires : un favorable à la station i et l'autre défavorable à la station j sur le même slot
- $c(i, j)$ quand il y a eu une collision entre un paquet de la station i et un paquet de la station j sur le même slot.

Il y a 3 stations et le cycle est divisé en 6 slots attribués comme suit: 123123 (cela veut dire que la station 1 est propriétaire des slots 1 et 4, la station 2 propriétaire des slots 2 et 5 et la station 3 propriétaire des slots 3 et 6). La station 1 doit émettre 8 trames dès $T=1$ et La station 3 a à émettre 4 trames dès $T=2$.

NB. Toutes les réponses doivent être justifiées. On tiendra compte de la présentation.

EXAMEN 2^{ème} session
INTELLIGENCE ARTIFICIELLE

Durée : 2 heures

Documents autorisés : Polycopiés, cours et travaux dirigés.

Remarques générales

**Vous ne devez joindre aucune feuille de l'énoncé de l'examen à votre copie.
Il est interdit d'écrire au crayon de bois.**

Partie Résolution de problème

Exercice 1 (4 points)

Sur une table sont posés trois cubes, les cubes A et B sur la table et le cube C sur le cube A.
On a trois "abscisses" possibles pour les cubes et celles-ci sont indifférenciées : la situation de la figure 2, par exemple, est « équivalente » à celle de la figure 1.

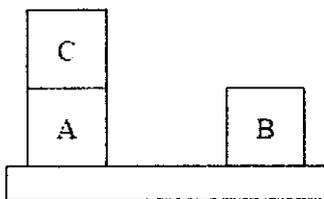


figure 1

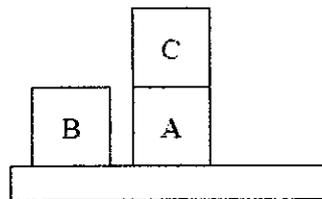


figure 2

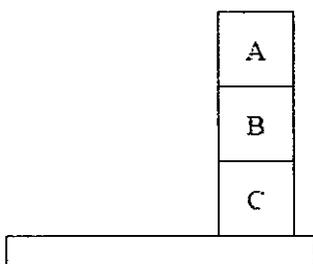


figure 3

On veut atteindre la situation d'arrivée présentée par la figure 3. On a juste le droit de soulever un cube qui n'est pas recouvert par un autre cube et de le reposer ailleurs. Le coût de cet opérateur est de 1.

Questions

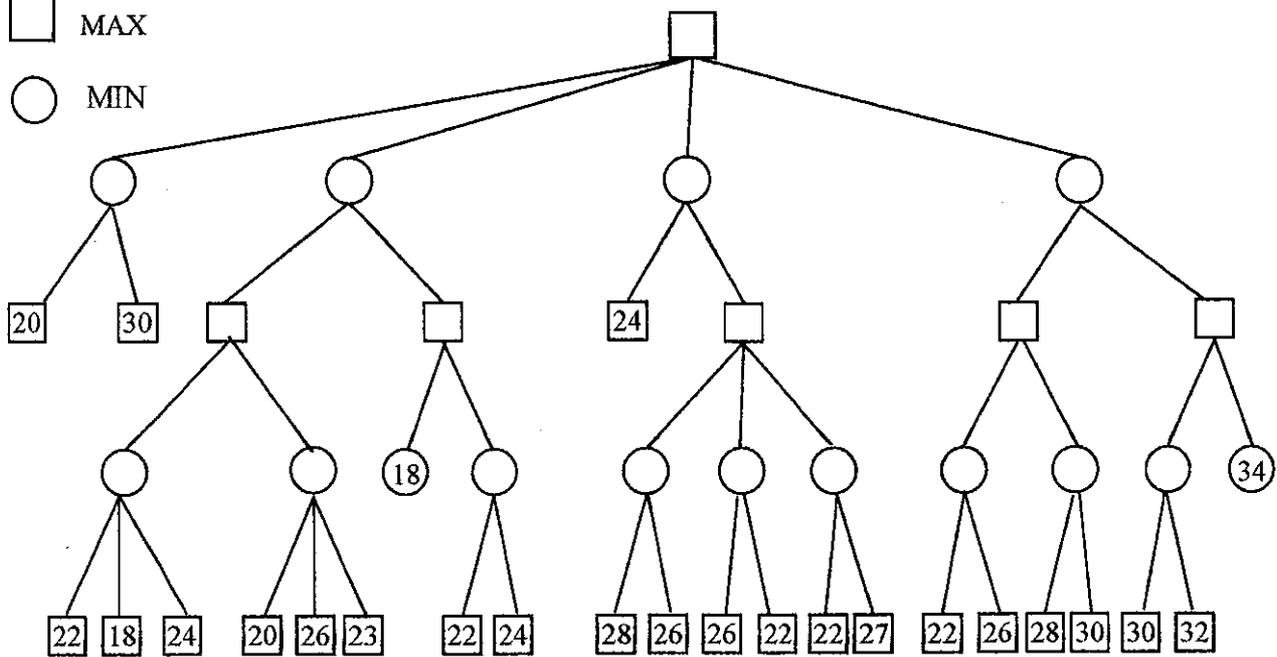
1. Nous considérons la fonction d'évaluation suivante
 $f(s) = -$ nombre d'abscisses occupées par un cube
Appliquer la méthode du gradient.
2. Nous considérons la fonction heuristique suivante
 $h(s) =$ nombre d'abscisses occupées par un cube
Appliquer l'algorithme A. La fonction h est-elle minorante ?

Exercice 2 (6 points)

Nous considérons l'arbre de jeu suivant :

□ MAX

○ MIN



Questions

1. Explorer l'arbre de jeu ci-dessus selon l'algorithme MinMax avec coupes $\alpha\beta$ (y compris les coupes profondes). Vous indiquerez pour chaque nœud la valeur remontée ainsi que la valeur α ou la valeur β (selon le cas).
2. Appliquer l'algorithme MTD (Memory test driver) sur l'arbre ci-dessus avec pour valeur $init_g + \infty$.

Partie Réseau de neurones

1- Fonction NAND (3 points)

Proposez un réseau de type Perceptron à seuil qui calcule la fonction NAND à trois entrées¹. Il faut préciser la structure du réseau, la valeur du seuil et la valeur des poids.

2- Apprentissage (7 points)

On veut faire apprendre à un Perceptron à seuil $\theta=0$ les exemples suivants :

| e1 | e2 | e3 | e4 | sortie |
|----|----|----|----|--------|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |

2.1. Faites tourner à la main l'algorithme d'apprentissage de Rosenblatt en mode batch (hors ligne), en détaillant les valeurs des poids à chaque pas d'apprentissage. On peut prendre comme poids initiaux $(w_1, w_2, w_3, w_4) = (0.1, 0.1, 0.1, 0.1)$ et comme coefficient d'apprentissage 0.1.

2.2. Même question en mode stochastique (en ligne).

¹ $\text{nand}(x,y,z) = \text{non}(\text{et}(x,y,z))$

