

جامعة الشهيد حمه لخضر - الوادي -



كلية العلوم الدقيقة
قسم الإعلام الآلي

الأسبنة الثنائية إعلام آلي

الأسبنة + الحلول النموذجية للامتحانات

السداسي الأول للموسم الجامعي 2016/2017

- الدورة العادية -

جامعة الشهيد حمه لخضر - الوادي -



كلية العلوم الدقيقة
قسم الإعلام الآلي

السنة الثانية إعلام آلي

السداسي الثالث - الدورة العادية -

السنة الجامعية 2017/2016



مقاييس امتحانات سنة ثانية إعلام آلي - السداسي الثالث -

الأستاذ : يعقوب محمد أمين	Logique Math
الأستاذ : عثمان بن سحر	POO
الأستاذة : بليلة خولة	Théorie de LANG
الأستاذ : خلايفة عبد الناصر	Anglais
الأستاذ : بن علي عبد الكامل	ALGO 1
الأستاذ : غربي قدور	Architecture Ord
الأستاذ : مديح الساسي	Système d'Info

Algorithmique et Structures de données

Examen du 08 janvier 2017

Durée 1h30 — Sans documents

Exercice 1 — Tri et Complexité

{6 points}

On peut décrire le tri par insertion de la façon suivante :

- On fait n itérations.
- Après l'itération $(i - 1)$ les premiers $(i - 1)$ éléments du tableau sont triés.
- A l'itération i on insère d'une manière séquentielle le i -ème élément du tableau parmi les premiers $(i - 1)$ éléments.

- (a) Illustrer le tri par insertion en indiquant le résultat de chaque itération de celui-ci à partir du tableau suivant : $t = \{5, 2, 4, 6, 1, 3\}$. (1)
- (b) Quels sont les nombres minimum et maximum de déplacements d'éléments que cet algorithme peut faire lors du tri d'un tableau de n éléments. (2)
- (c) Réaliser en C la procédure TRI_INSERT qui implémente cet algorithme. (3)

Exercice 2 — Utilisation des piles et files

{8 points}

- (a) Soit P une pile d'entiers, quel est l'état de la pile après les opérations suivantes :
PILEVIDE(P) ; EMPILER(11, P) ; EMPILER(7, P) ; EMPILER(3, P) ; DÉPILER(P) ;
EMPLER(SOMMET(P), P) ; EMPILER(9, P) ; DÉPILER(P) ; DÉPILER(P) ; (1)
- (b) Réaliser les opérations EMPILER et DEPILER pour une pile implémentée en utilisant un tableau. Commencer par définir le type de données. (3)
- (c) Soit F une file d'entiers, quel est l'état de la file après les opérations suivantes :
FILEVIDE(F) ; ENFILER(17, F) ; ENFILER(9, F) ; ENFILER(TETE(F), F) ; DEFILER(F) ;
ENFILER(6, F) ; DEFILER(F) ; (3)
- (d) Réaliser les opérations EMPILER et DEFILER pour une file implémentée en utilisant un tableau circulaire. Commencer par définir le type de données. (3)

Exercice 3 — Interclassement de listes triées

{6 points}

Réaliser en C une fonction pour interclasser deux listes triées dans une troisième liste triée. Les listes sont implémentées en utilisant des listes simplement chaînées. Commencer par définir les types de données et réaliser toute autre fonction dont on a besoin.

Corrigé de l'examen du 08 janvier 2017
Algorithmique et Structures de données
Licence 2 d'informatique

Exercice 01 : Tri et Complexité

(6 points)

(a) Illustration de l'algorithmique de tri par insertion à partir du tableau $t = \{5, 2, 4, 6, 1, 3\}$

(1 pt)

Input	: 5 2 4 6 1 3
Itération 1	: 5 2 4 6 1 3
Itération 2	: 2 5 4 6 1 3
Itération 3	: 2 4 5 6 1 3
Itération 4	: 2 4 5 6 1 3
Itération 5	: 1 2 4 5 6 3
Itération 6	: 1 2 3 4 5 6
Output	: 1 2 3 4 5 6

(b) Nombres minimum et maximum de déplacements d'éléments que fait cet algorithme sur un tableau de n éléments

(2 pts)

- L'algorithme de tri par insertion fait au maximum $\sum_{i=1}^{n-1} i = n(n-1)/2$ déplacements d'éléments ; c'est le cas où le tableau est trié dans l'ordre inverse. (1 point)
- Il ne fait aucun déplacement si le tableau est déjà trié. (1 point)

(c) Réalisation en C de l'algorithme de tri par insertion

(3 pts)

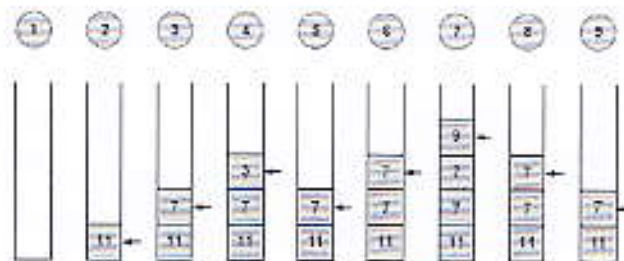
```
void triInsertion (float t[])
{
    int i, k ;
    float x ;
    for (i = 1; i < n; ++i)
    {
        k = i - 1;
        x = t[i];
        while ((k >= 0) && (t[k] > x))
            t[k + 1] = t[k], k --;
        t[k + 1] = x;
    }
}
```

Exercice 02 : Piles et Files

(8 points)

(a) Etat de la pile

(1 pt)



(b) Réalisation des opérations EMPILER et DEPILER

(3 pts)

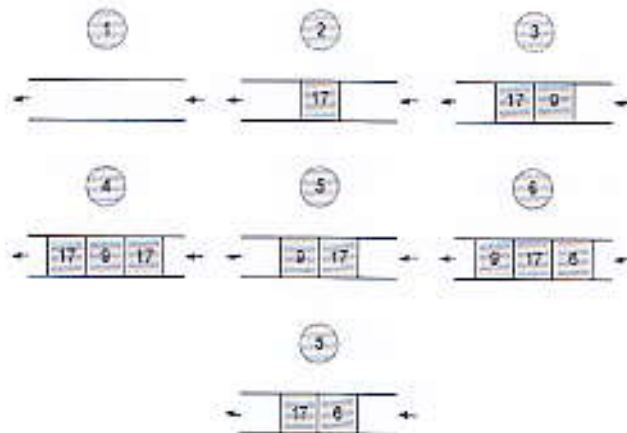
```
#define nMax 100
/* Déclaration des types */
typedef int element;
typedef struct Pile
{
    element contenu[nMax]; /* tableau qui va contenir les éléments
                           de la pile */
    int hauteur; /* indice du sommet de la pile */
} Pile;

/* EMPILER ; la pile doit être non pleine */
void EMPILER(Pile *p, element x)
{
    p->contenu[p->hauteur++] = x;
}

/* DEPILER ; la pile doit être non vide */
void DEPILER(Pile *p)
{
    p->hauteur--;
}
```

(c) Etat de la file

(1 pt)



(d) Réalisation des opérations EMFILER et DEFILER

(3 pts)

```
#define nMax 100
/* Déclaration des types */
typedef int element;
typedef struct File
{
    element contenu[nMax]; /* tableau qui va contenir les éléments
                           de la file */
    int tete, queue; /* indice du premier élément et l'indice de la
                     dernière case disponible après le dernier
                     élément */
} File;

/* EMFILER ; la file doit être non pleine */
void EMFILER(File *f, element x)
{
    f->contenu[f->queue++] = x;
    if (f->queue == nMax)
        f->queue = 0;
}
```

```

/* DEFILER ; la file doit être non vide */
void DEFILER(File *f)
{
    f->tete--;
    if (f->tete == nMax)
        f->tete = 0;
}

```

Exercice 03 : interclassement de listes triées

(6 points)

(a) Déclaration de types

```

/* Déclaration des types */
typedef float element ;
struct Cellule
{
    element contenu;
    struct cellule *suivant;
};
typedef struct Cellule Cellule, *Liste ;

```

(b) Fonction en question

(4 pts)

```

/* Fonction qui interclasse de deux listes triées */
/* Elle retourne un pointer vers la liste résultat */
Liste Interclassement (Liste a, Liste b)
{
    Liste c;

    while (!EstListeVide(a) && !EstListeVide(b))
    {
        if (tete(a) < tete(b))
        {
            c = ajouter_fin(tete(a), c);
            a = a->suivant;
        }
        else
        {
            c = ajouter_fin(tete(b), c);
            b = b->suivant;
        }
    }

    if (!EstListeVide(a))
        while (!EstListeVide(a))
        {
            c = ajouter_fin(tete(a), c);
            a = a->suivant;
        }
    if (!EstListeVide(b))
        while (!EstListeVide(b))
        {
            c = ajouter_fin(tete(b), c);
            b = b->suivant;
        }

    return c ;
}

```

(c) Fonction pour tester si une liste est vide

```
int EstListeVide (Liste a)
{
    return (a == NULL) ;
}
```

(d) Fonction retourner l'élément en tête d'une liste

```
/* La liste doit être non vide */
element tete (Liste a)
{
    return a->contneu ;
}
```

(e) Fonction pour ajouter un nouvel élément à la fin d'une liste

(2 pts)

```
/* Ajout à la fin d'une liste */
Liste ajouter_fin (Liste a, element x)
{
    Liste nou_ele, b ;

    nou_ele = (Liste) malloc (sizeof(Cellule)) ;
    nou_ele->contenu = x ;
    nou_ele->suivant = NULL ;

    if (EstListeVide(a))
        return nou_ele ;
    b = a ;
    while (!EstListeVide(b->suivant))
        b = b->suivant ;
    b->suivant = nou_ele ;
    return a ;
}
```


Examen du 3^{ème} semestre

Exercice 01 (04pts):

Trouvez le type de chacune des grammaires suivantes selon la classification de Chomsky :

- 1- $G = (\{a,b\}, \{A,B,S\}, S, R : S \rightarrow Sa/Sb/\epsilon)$.
- 2- $G = (\{a,b\}, \{A,B,S\}, S, R : S \rightarrow AB, A \rightarrow aA/\epsilon, B \rightarrow bB/\epsilon)$.
- 3- $G = (\{a,b,c\}, \{S,T\}, S, R : S \rightarrow Ta/Sa, T \rightarrow Tb/Sb/\epsilon)$.
- 4- $G = (\{0,1,2\}, \{S,C,Z,T\}, S, \{S \rightarrow TZ, T \rightarrow 0T1C/\epsilon, C1 \rightarrow 1C, CZ \rightarrow Z2, 1Z \rightarrow 1\})$.

Exercice 02 (04pts):

Trouver les grammaires qui engendrent les langages suivants :

- Tout les mots sur $\{a,b\}$ qui contiennent le facteur ab .
- Tout les nombres binaires.

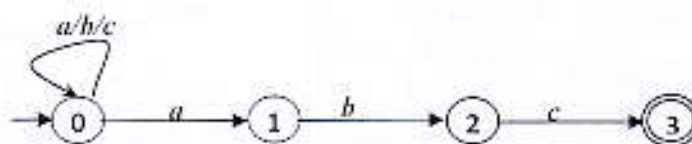
Exercice 03 (06pts):

Trouvez l'automate équivalent à l'expression régulière r selon la méthode des dérivées tel que :

$$r = (b(a+b+c)^* + c(a+b+c)^*)^*$$

Exercice 04 (06pts):

Soit A l'automate ci-dessous :



- Trouvez A_{Det} : l'automate déterministe équivalent à A ?
- Trouvez A_C : l'automate du langage inverse (complémentaire) de A_{Det} ?
- Quel langage représente l'automate A_C ?

Bon Courage

ex 01

1) $s \rightarrow sa$ (type 3)
 $s \rightarrow sb$ (type 3)
 $s \rightarrow \epsilon$ (type 3) } type 3 (1 pts)

2) $s \rightarrow AB$ (type 2)
 $A \rightarrow aA$ (type 3)
 $A \rightarrow aA/\epsilon$ (type 3)
 $B \rightarrow bB/\epsilon$ (type 3) } type 2 (1 pts)

3) + 4) (voir le TD) (1 pts) + (1 pt)

ex 02

1) $G = (\{a, b\}, \{s\}, s, R: s \rightarrow as/bs/ab)$ (02 pts)

2) $G = (\{0, 1\}, \{s\}, s, R: s \rightarrow os/1s/01)$ (02 pts)

ex 03

$r = (b(a+bc)^* + c(a+bc)^*)^*$

$r // a = \emptyset$ (0,5)
 $r // b = (a+bc)^* r$ (0,5)
 $r // c = (a+bc)^* r$ (0,5)

$(a+bc)^* r // a = (a+bc)^* r$ (0,5)
 $(a+bc)^* r // b = (a+bc)^* r$ (0,5)
 $(a+bc)^* r // c = (a+bc)^* r$ (0,5)

$q_0 = r, q_1 = (a+bc)^* r$
 q_0 et q_1 sont les deux états finaux et q_0 est l'état initial

l'automate = (1 pt)

	a	b	c
q_0	q_1	q_1	q_1
q_1	q_1	q_1	q_1

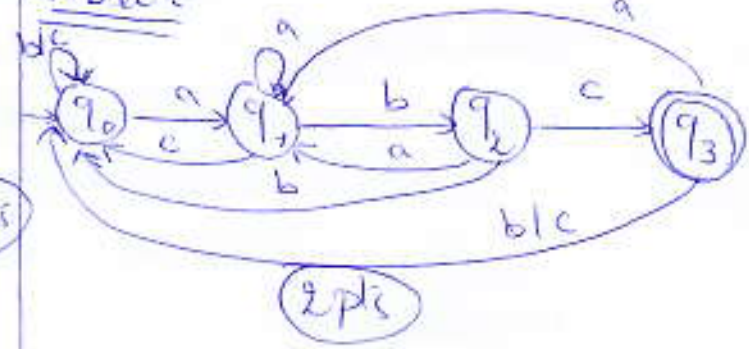


ex 03

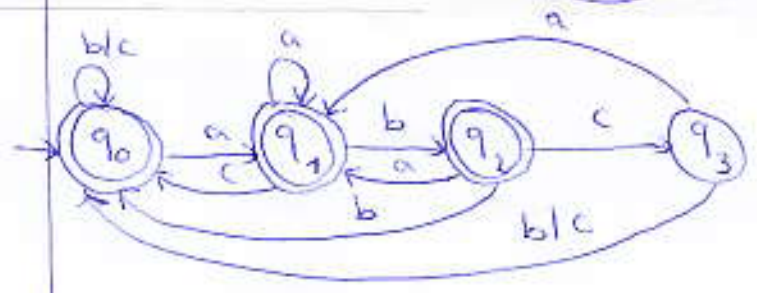
	a	b	c
0	{0,1}	0	0
{0,1}	{0,1}	{0,2}	0
{0,2}	{0,1}	0	{0,3}
{0,3}	{0,1}	0	0

$q_0 = 0, q_1 = \{0,1\}, q_2 = \{0,2\}, q_3 = \{0,3\}$
 q_0 : l'état initial et q_3 : état final

A det



Ac (complémentaire) (1 pt)



3) - l'automate A_c représente le langage de tout les mots sur $\{a, b, c\}$ qui ne se terminent pas par le facteur abc (1 pts)

(ن) س1- ما الفرق بين (class abstraite) و (interface) (أذكر اثنين فقط)
 (ن) س2- هناك خطأ في البرنامج التالي؟ ما هما؟ ولماذا؟ قم بتعديل البرنامج ليصبح قابلا للتنفيذ

```

1 interface C {
2     public int m();
3     public abstract int n();
4 }
5
6 abstract class D implements C {
7     public int m() {
8         return 3;
9     }
10 }
11
12 abstract class E extends D {
13     public int n() {
14         return 4;
15     }
16     public int o() {
17         return 5;
18     }
19 }
    
```

```

19 class F extends E {
20 }
21 public class Problem3 {
22     public static void main(String [] args) {
23         F f = new F();
24         f.m();
25         f.n();
26         f.o();
27         E e = new E();
28         e.m();
29         e.n();
30         e.o();
31         C c = new F();
32         c.m();
33         c.n();
34         c.o();
35     }
36 }
    
```

(ن) س3- هل البرنامج التالي قابل للتنفيذ؟ إذا كانت اجابتك نعم ما هي نتيجة البرنامج؟ إذا كانت اجابتك لا صحح الأخطاء ليصبح قابلا للتنفيذ (ملاحظة: في حالة التصحيح غير مسموح بحذف أي سطر أو جعله تعلق)

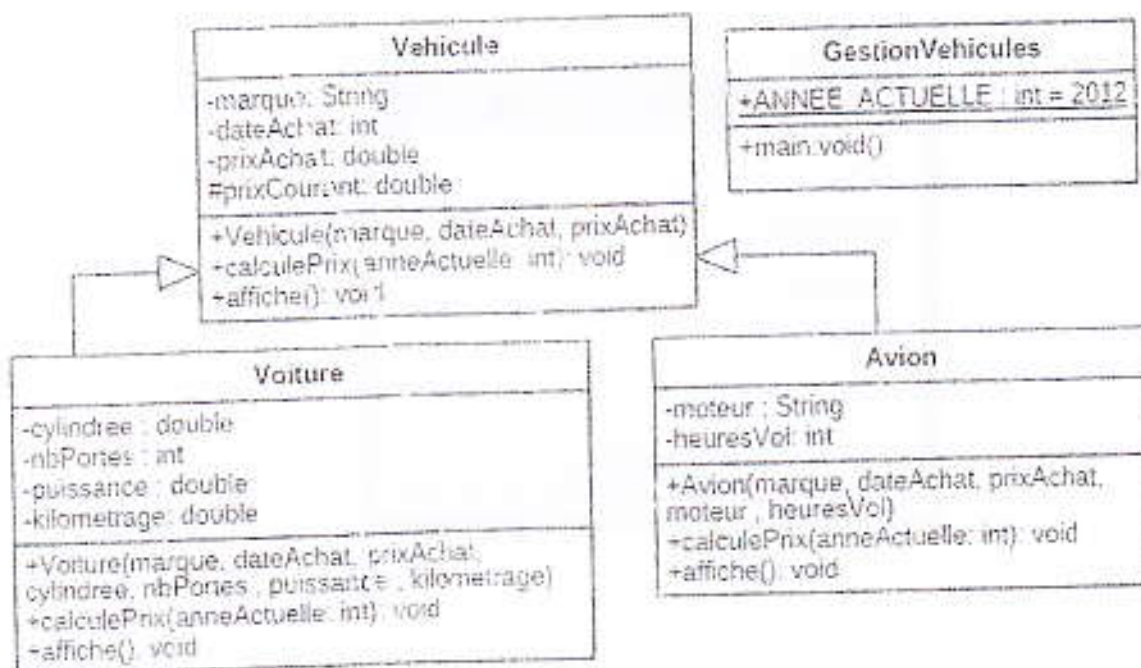
```

1 public class Question1 extends A implements B, C {
2     public void bar() {
3         System.out.println("Q1");
4     }
5     public void foo() {
6         System.out.println("Q2");
7     }
8     public static void main(String [] args) {
9         A a = new Question1();
10        a.bar(); a.foo();
11        A a1 = new A();
12        a1.bar(); a1.foo();
13        B b = new Question1();
14        b.bar(); b.foo();
15        C c = new A();
16        c.foo();
17    }
18 }
    
```

```

19 class A implements B {
20     public void bar() {
21         System.out.println("A1");
22     }
23     public void foo() {
24         System.out.println("A2");
25     }
26 }
27
28 interface B extends C {
29     public void bar();
30 }
31
32 interface C {
33     public void foo();
34 }
    
```

(ن) تمرين 1 لدينا المعطى التالي



1- عرف جميع الفئات باستخدام لغة الجافا بحيث
 -الدالة affiche تقوم بطباعة جميع خصائص الفئة
 -الدالة calculePrix تحسب السعر الحالي كالتالي

السعر الحالي = سعر الشراء * (1-%1) * (النسبة الحالية - سنة الشراء)	Vehicule
السعر الحالي = سعر الشراء * (1-%1) * (النسبة الحالية - سنة الشراء) + %5 عدد الكيلومترات المقطوعة / (1000-%1) اذا كانت السيارة من ماركة "Porsche" او "Ferrari" و الا %2 اذا كانت من ماركة اخرى	Voiture
السعر الحالي = سعر الشراء * (1-%1) * (عدد ساعات التحليق / 100) اذا كان المحرك من النوع "HELICES" و الا عدد ساعات التحليق / (1000-%1)	Avion
ملاحظة:السعر الحالي دائما موجب أي اذا كان السعر الحالي بعد الحساب سالبا فان السعر الحالي بأخذ القيمة صفر	

2- أكمل الدالة الرئيسية بحيث تعطي نتيجة التنفيذ التالية

GestionVehicules	نتيجة التنفيذ
<pre> class GestionVehicules { private static int ANNEE_ACTUELLE = 2012; public static void main(String[] args) { Voiture[] garage = new Voiture[3]; Avion[] hangar = new Avion[2]; for (int i = 0; i < garage.length; i++) { garage[i].calculePrix(ANNEE_ACTUELLE); garage[i].affiche(); } for (int i = 0; i < hangar.length; i++) { hangar[i].calculePrix(ANNEE_ACTUELLE); hangar[i].affiche(); } } </pre>	<pre> ---- Voiture ---- marque : Peugeot, date d'achat : 2015, prix actuel : 119333.8 2.5 litres, 5 portes, 186.0 CV, 12000.0 km. ---- Voiture ---- marque : Porsche, date d'achat : 1999, prix actuel : 135000.0 ,5.5 litres, 2 portes, 280.0 CV, 81320.0 km. ---- Voiture ---- marque : Fiat, date d'achat : 2001, prix actuel : 4982.5 1.6 litres, 3 portes, 65.0 CV, 3000.0 km. ---- Avion à HELICES ---- marque : Cessna, date d'achat : 1982, prix actuel : 923005.4 250 heures de vol ---- Avion à REACTION ---- marque : Nain Connu, date d'achat : 1993, prix actuel : 3759355.26 1300 heures de vol. </pre>

3- اقترح تحسينا للدالة الرئيسية
 (6ن) تمرين 2- باستخدام (java.awt) أكتب تطبيقا يظهر في نافذته العناصر المبينة في الشكل بحيث يسمح للمستخدم إضافة سيارة إلى الجدول garage

(ن2) س1- ما الفرق بين (interface) و (class abstraite) (اذكر اثنين فقط)
 (ن2) س2- هناك خطأ في البرنامج التالي؟ ما هما؟ ولماذا؟ قم بتعديل البرنامج ليصبح قابلا للتنفيذ

```

1 interface C {
2     public int m();
3     public abstract int n();
4 }
5 abstract class D implements C {
6     public int m() {
7         return 3;
8     }
9 }
10 abstract class E extends D {
11     public int n() {
12         return 4;
13     }
14     public int o() {
15         return 5;
16     }
17 }
18
    
```

```

19 class F extends E {
20 }
21 public class Problem3 {
22     public static void main(String [] args) {
23         F f = new F();
24         f.m();
25         f.n();
26         f.o();
27         E e = new E();
28         e.m();
29         e.n();
30         e.o();
31         C c = new F();
32         c.m();
33         c.n();
34         c.o();
35     }
36 }
    
```

(ن2) س3- هل البرنامج التالي قابل للتنفيذ؟ إذا كانت اجابتك نعم ما هي نتيجة البرنامج؟ إذا كانت اجابتك لا صحح الأخطاء ليصبح قابلا للتنفيذ (ملاحظة: في حالة التصحيح غير مسموح بتدقيق أي سطر أو جعله تعليقا)

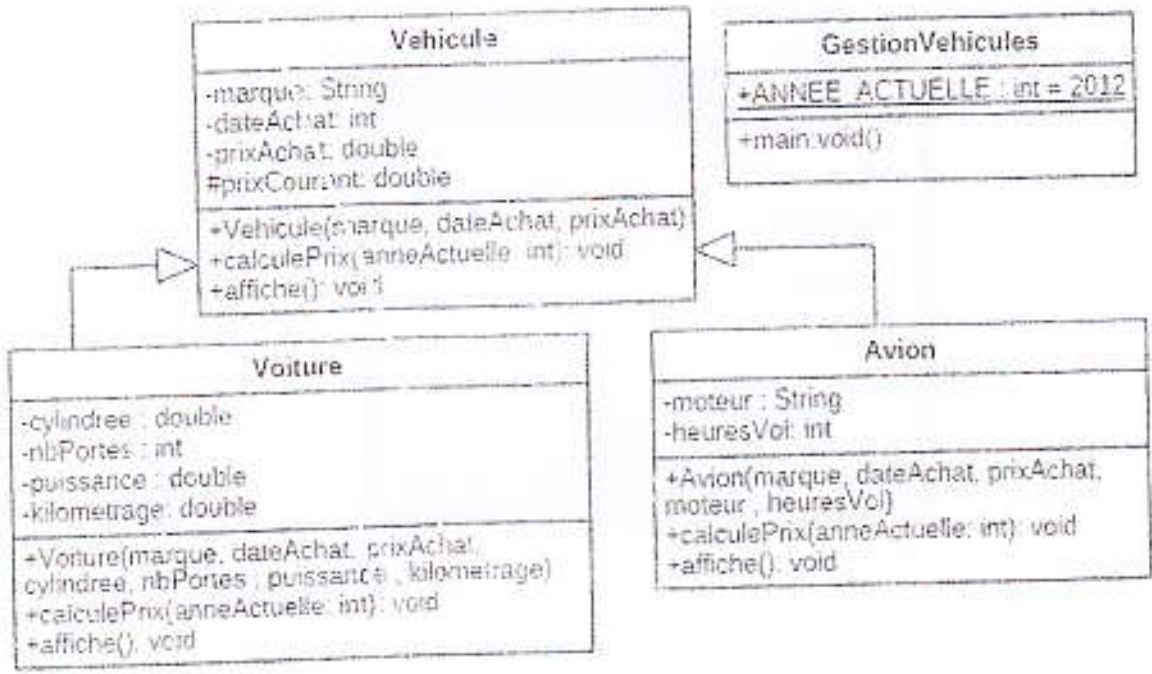
```

1 public class Question1 extends A implements B, C {
2     public void bar() {
3         System.out.println("Q1");
4     }
5     public void foo() {
6         System.out.println("Q2");
7     }
8     public static void main(String [] args) {
9         A a = new Question1();
10        a.bar(); a.foo();
11        A a1 = new A();
12        a1.bar(); a1.foo();
13        B b = new Question1();
14        b.bar(); b.foo();
15        C c = new A();
16        c.foo();
17    }
18 }
    
```

```

19 class A implements B {
20     public void bar() {
21         System.out.println("A1");
22     }
23     public void foo() {
24         System.out.println("A2");
25     }
26 }
27
28 interface B extends C {
29     public void bar();
30 }
31
32 interface C {
33     public void foo();
34 }
    
```

(ن8) تمرين 1 لدينا المخطط التالي



- 1-Au moins une méthode d'une classe abstraite est abstraite
 2-Toutes les méthodes d'une interface sont abstraites

1-0.5% The method o is undefined for class C. The only methods that can be called on an object are those which are defined in the compile-time type of the object.

Fixes (0.5% for one of the following)

1. We could add an abstract method o into interface C.
2. We could change line 35 to say E e = new F();

2-0.5% E is an abstract class and therefore can't be instantiated.

Fix (0.5%)

The line could be changed to:

E e = new F();

0.5% - Yes, the code compiles

0.5% - for correct output

Q1

Q2

A1

A2

Q1

Q2

A2

```

class GestionVehicules {
    // Pour représenter l'année courante;
    // Il est aussi possible d'utiliser la classe Date
    // pour récupérer cette information automatiquement
    private static int ANNEE_ACTUELLE = 2012;
    public static void main(String[] args) {
        Vehicule[] vehicules = new Vehicule [5];
        vehicules[0] = new Voiture("Peugeot", 2005, 147325.79, 2.5, 5, 180.0, 12000);
        vehicules[1] = new Voiture("Porsche", 1999, 250000.00, 6.5, 2, 280.0, 81320);
        vehicules[2] = new Voiture("Fiat", 2001, 7327.30, 1.6, 3, 65.0, 3000);
        vehicules[3] = new Avion("Cessna", 1982, 1230673.90, "HELICES", 250);
        vehicules[4] = new Avion("Main Connu", 1993, 4321098.00, "REACTION", 1300);
        for (int i = 0; i < vehicules.length; i++) {
            vehicules[i].calculePrix(ANNEE_ACTUELLE);
            vehicules[i].affiche();
        }
    }
}

class Vehicule {
    private String marque;
    private int dateAchat;
    private double prixAchat;
    private double prixCourant;
    public Vehicule(String marque, int date, double prix) {
        this.marque = marque;
        dateAchat = date;
        prixAchat = prix;
        prixCourant = prix;
    }
    public void calculePrix(int anneeActuelle) {
        double decote = (anneeActuelle - dateAchat) * 0.01;
        prixCourant = Math.max(0.0, (1.0 - decote) * prixAchat);
    }
    public void affiche() {
        System.out.print("marque : " + marque + ", date d'achat : " + dateAchat + ",
prix actuel : " + prixCourant);
        System.out.println();
    }
    public int getDateAchat(){
        return dateAchat;
    }
    public String getMarque(){
        return marque;
    }
    public double getPrixAchat(){

```

```

return prixAchat;
}
public void setPrixCourant(double prix){
    prixCourant = prix;
}
}
class Voiture extends Vehicule {
    private double cylindree;
    private int nbPortes;
    private double puissance;
    private double kilometrage;
    public Voiture(String marque, int date, double prix, double cylindree, int portes,
double cv, double km) {
        super(marque, date, prix);
        this.cylindree = cylindree;
        nbPortes = portes;
        puissance = cv;
        kilometrage = km;
    }
    public void calculePrix(int anneeActuelle) {
        double decote = (anneeActuelle - getDateAchat()) * .02;
        decote += 0.05 * (int)(kilometrage / 10000);
        if ((getMarque() == "Fiat") || (getMarque() == "Renault")) {
            decote += 0.1;
        } else if ((getMarque() == "Ferrari") || (getMarque() == "Porsche")) {
            decote -= 0.2;
        }
        setPrixCourant(Math.max(0.0, (1.0 - decote) * getPrixAchat()));
    }
    public void affiche() {
        System.out.println(" ---- Voiture ----");
        super.affiche();
        System.out.println(cylindree + " litres, " + nbPortes + " portes, " +
puissance + " CV, " + kilometrage + " km.");
    }
}
class Avion extends Vehicule {
    private String moteur;
    private int heuresVol;
    public Avion(String marque, int date, double prix, String moteur, int heures) {
        super(marque, date, prix);
        this.moteur = moteur;
        heuresVol = heures;
    }
    public void calculePrix(int anneeActuelle) {
        double decote;
        if (moteur == "HELICES") {
            decote = 0.1 * heuresVol / 100.0;
        } else {
            decote = 0.1 * heuresVol / 1000.0;
        }
        setPrixCourant(Math.max(0.0, (1.0 - decote) * getPrixAchat()));
    }
    public void affiche() {
        System.out.println(" ---- Avion à " + moteur + " ----");
        super.affiche();
        System.out.println(heuresVol + " heures de vol.");
    }
}
-----
import java.awt.*;
import java.awt.event.*;

public class Avion extends Frame implements ActionListener {
    Avion[] hangar;
    int index=0;
}

```

```

Label lmarque;
Label ldatach;
Label lpriach;
Label lmoteur;
Label lhervol;
TextField tfmarque;
TextField tfdatach;
TextField tfpriach;
TextField tfmoteur;
TextField tfhervol;
Button bAjouter;
Panel p;
AwtAvion(Avion[] Tab,int i){
super("Ajouter Avion");
hangar=Tab;
index=i;

    lmarque = new Label("Marque");
    ldatach = new Label("Date d'achat");
    lpriach = new Label("Prix d'achat");
    lmoteur = new Label("Moteur");
    lhervol = new Label("Heures de vol");
    tfmarque = new TextField();
    tfdatach = new TextField();
    tfpriach = new TextField();
    tfmoteur = new TextField();
    tfhervol = new TextField();
    bAjouter = new Button("Ajouter");
    bAjouter.addActionListener(this);
    GridLayout gl = new GridLayout(5,2);
    p=new Panel();
    p.setLayout(gl);
    p.add(lmarque);
    p.add(tfmarque);
    p.add(ldatach);
    p.add(tfdatach);
    p.add(lpriach);
    p.add(tfpriach);
    p.add(lmoteur);
    p.add(tfmoteur);
    p.add(lhervol);
    p.add(tfhervol);
    this.add(bAjouter, BorderLayout.SOUTH);
    this.add(p, BorderLayout.CENTER);

pack();
setVisible(true);
}

@Override
public void actionPerformed(ActionEvent arg0) {
// TODO Auto-generated method stub
hangar[index]= new Avion(tfmarque.getText(), tfdatach.getText(), tfpriach.getText(), tfmoteur.getText(), tfhervol.getText());
}
}

```


Examen de logique mathématique (2017)

Exercice 1 (10):

Mon fils a des tendances sportives. Et je veux l'aider à choisir le bon sport pour lui. J'ai des indications sous la forme de la devinette suivante :

ابني له ميولات رياضية. اردت مساعدته لاختيار الرياضة المناسبة له. لدي معطيات على شكل الاحجية الآتية:

- 1) S'il déteste (بكره) le football (كرة القدم), alors il aime (يحب) le tennis (التنس) ou le handball (كرة اليد).
- 2) Il a les mêmes préférences (له نفس التفضيلات) pour le tennis et la natation (السباحة).
- 3) S'il déteste le handball alors il aime le tennis.
- 4) Il déteste le football ou le handball, ou les deux autres.
- 5) S'il aime le handball et pas le football, alors il aime la natation.

Après longue réflexion, je décide de l'introduire à un club de football, car, selon une analyse :

بعد تفكير طويل قررت إدخاله لنادي كرة قدم، لأنه، تبعاً لتحليل نادي النتيجة الآتية:

C) Il déteste la natation, le handball et le tennis.

Donnez une représentation logique de ce raisonnement et cherchez si cette conclusion C est une conséquence valide des 5 affirmations énoncées au-dessus.

هناك تمثيل منطقي لهذا التفكير. وبين اذا كانت النتيجة مستنبطة من المعطيات الخمسة بالاعلى.

ملاحظة: $Aimer = préférer = \neg détester$

Exercice 2 (6):

On se placera dans le domaine d'interprétation de l'ensemble des êtres humains (البشر) et on utilisera des prédicats pour exprimer Voleur(x) (سارق), Assassin(x) (سفاح), Criminel(x) (مجرم), et Famille(x,y)

- 1) Certaines personnes sont des assassins
- 2) Tout assassin a déjà volé
- 3) Les voleurs ne sont pas tous des assassins
- 4) Un criminel ne peut être qu'un assassin ou un voleur ou les deux.
- 5) Il y a un criminel dans ma famille
- 6) Tous les membres de ma famille sont des voleurs

Exercice 3 (4):

Utilisez quatre méthodes différentes pour montrer :

استعمل أربع طرق مختلفة لإثبات:

$$\vdash (A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$$

Bon courage

2017 L.M ۱۲۰۱

Exo 1:

- 1) A : ۱) détente de foot
 B : ۲) ۳) tennis
 C : ۴) ۵) handball
 D : ۶) ۷) ۸) natation

1) $A \rightarrow (B \vee C)$

2) $B \vee D$

3) $C \rightarrow B$

4) $A \vee C \vee (B \wedge D)$

5) $\neg C \wedge A \rightarrow B$

6) $B \wedge C \wedge D$

7) 1), 2), 4), 5), 6) $\vdash C$ non valide.

Exo 2:

1) $\exists x A(x)$

2) $\forall x (A(x) \rightarrow V(x))$

3) $\exists x (V(x) \wedge \neg A(x))$

4) $C(x) \rightarrow A(x) \vee V(x) \vee (A(x) \wedge V(x))$

5) $\exists x (C(x) \wedge F(x, y))$

6) $\forall x (F(x, y) \rightarrow V(x))$

Exo 3:

1) T.V

2) M.Pens $A \rightarrow B, A \rightarrow (B \wedge C), A \vdash C$

3) la résolution

4) Tableaux sémantiques.

Examen de semestre
10/01/2017

 **Documents non autorisés ; Durée ⌚ 01h30.**

Exercice 01 : Suivi des clients négligents (10 pts)

Une société travaille avec plus de 150 clients en comptes. En début de mois, le service "Comptabilité – Finances" demande un état des clients négligents (clients ayant plus de 5 jours de retard dans le paiement d'une facture) au service "Comptabilité-clients". Pour chacun de ces clients, une demande d'information est transmise au service commercial par le biais de la messagerie interne. Sur la base des informations recueillies, le service prend contact avec les clients pour leur faire part du retard et obtenir une explication. Si la situation financière du client ne lui apparaît pas trop critique, elle met en place de nouvelles conditions de paiement. Dans le cas contraire elle transmet le dossier au service contentieux qui démarrera une procédure judiciaire en recouvrement de créance.

1. Etablir le diagramme des flux. (5 Pts)
2. Etablir le modèle conceptuel de traitement M.C.T. (5 Pts)

Exercice 02 : Gestion d'une auto-école (5 pts)

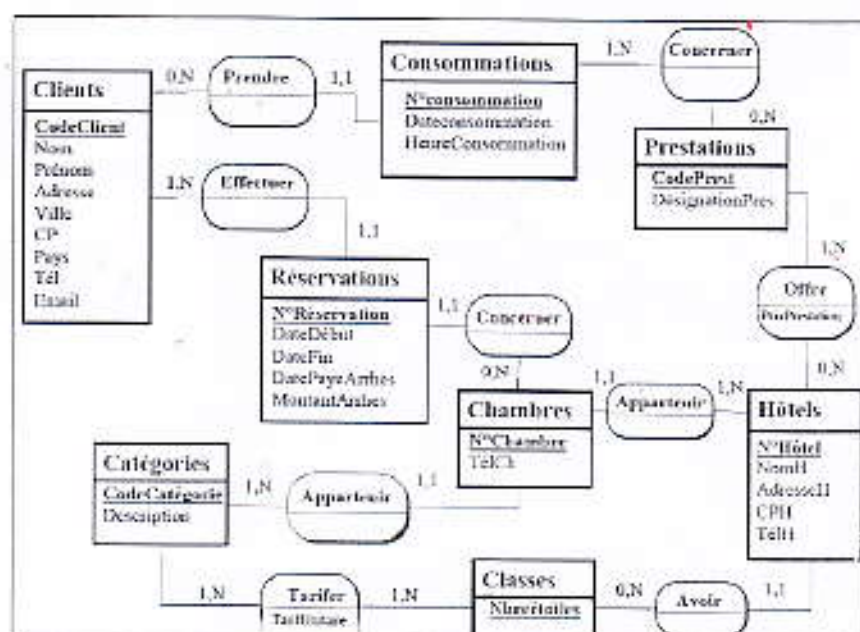
Une auto-école comprend des moniteurs et des véhicules. Elle prépare des élèves à passer un permis de type donné. Les moniteurs donnent des leçons aux élèves ; une leçon possède une durée (durlec). Les règles de gestion sont les suivantes :

- Une leçon n'est donnée que par un seul moniteur
- Un élève ne passe qu'un seul type de permis
- Une leçon est donnée que pour un seul élève
- Un élève n'apprend à conduire que sur un seul véhicule

Travail demandé : Etablir le modèle conceptuel des données correspondant.

Exercice 03 : (5 pts)

Soit le MCD suivant :



Travail demandé :

En appliquant les règles de passage au modèle relationnel, établir le Modèle logique des données.



The first exam of English

Read carefully the text and answer the questions

A computer system consists of two major elements: hardware and software. Computer hardware is the collection of all the parts you can physically touch. Computer software, on the other hand, is not something you can touch. Software is a set of instructions for a computer to perform specific operations. You need both hardware and software for a computer system to work.

The keyboard, the monitor, the mouse and the processing unit are the physical elements of a computer that means Hardware. However, most of a computer's hardware cannot be seen; in other words, it is not an external element of the computer, but rather an internal one, surrounded by the computer's casing. A computer's hardware is comprised of many different parts, but perhaps the most important of these is the motherboard. The motherboard is made up of even more parts that power and control the computer.

The functioning of the hardware is directed by a collection of routines, rules and symbolic languages that describe the Software. Software is capable of performing specific tasks, as opposed to hardware which only perform mechanical tasks that they are mechanically designed for. Practical computer systems divide software systems into three major groups: System software, Programming software and Application software.

Questions

1- Give the basic ideas of each paragraph. (3 pts)

- 1-.....
- 2-.....
- 3-.....

2- Give the main idea for this text. (1 pts)

.....

3- Answer the following questions from the text (2 pts)

- What are the parts of computer that you cannot physically touch?

.....

- Which parts of computer can perform mechanical tasks?

.....

4- Match the terms in the box with the explanations or definitions below: (4 pts)

-internet	- hacker	- firewall	- CPU	- input
-web	-output	- Hard drive		

- a- The primary means of storage, it can hold a massive amount of information.....
- b- Software which blocks attempts by others to access your computer over the internet.....
- c- a global network of interconnected computers.....
- d- The information which is presented to the computer.....
- e- Results produced by a computer.....
- f- A person who illegally accesses somebody else's computer over the internet.....
- g- The billions of pages those are stored on large computers.....
- h- The brain of the computer.....

5- Extract from the text the word or sentence that has the same meaning as these words (2 pts)

CPU....., development.....

Inside....., screen.....

6- Give from the text the english translation of these words: (4 pts)

Carte mère....., unité de traitement....., écran.....

ensemble....., clavier....., tâche.....

7- Write the verbs between brackets in the correct form (2 pts)

- Don't shut down the computer, it (to scan)the USB drive for virus.
- The operating system (to contain)the information that runs all the system's basic tasks.
- Normally, my modem (to connect)..... to internet quickly. But this days it (to take)a few minutes to establish the connection.

8- Write the following words into the spaces. (3 pts)

request - traffic - modem - roads - functions - server.

To **access** the web, you need a computer and a (1)..... You then connect over your telephone line to an internet service provider (ISP), which sends your (2) to view a particular web page to the correct web (3)..... Websites are not the only service available on the internet. It is also used for many other (4)..... including sending and receiving email, and connecting to newsgroups and discussion groups. You could say that the internet is a system of (5)....., and web pages and emails are types of (6)..... that travel on those roads.

The correction of the first exam in English

1- The basic ideas of each paragraph. (3 pts)

- The major elements of computer system.
- Definition of hardware.
- Definition of software.

2- Title for this text. (1 pts)

Composition of the computer.

3- Answers of the questions from the text (2 pts)

- The parts of computer that you cannot physically touch are Software.
- The parts of computer that can perform mechanical tasks are Hardware.

4- Terms and definitions (4 pts)

- a- The primary means of storage, it can hold a massive amount of information **Hard drive**
- b- Software which blocks attempts by others to access your computer over the internet **firewall**
- c- a global network of interconnected computers **internet**
- d- The information which is presented to the computer **input**
- e- Results produced by a computer **output**
- f- A person who illegally accesses somebody else's computer over the internet **hacker**
- g- The billions of pages those are stored on large computers **web**
- h- The brain of the computer **CPU**.

5- The synonyms (2 pts)

CPU = The processing unit. Development = Programming.

Inside = internal. Screen = Monitor.

6- The english translation (3 pts)

Carte mère = motherboard. Unité de traitement = processing unit.

Ecran = monitor. Ensemble = set.

Clavier keyboard. Tâche = task..

7- The correct form of the verbs (2 pts)

- Is scanning - contains - connects - is taking.

8- Words in the spaces. (3 pts)

1. modem.
2. request.
3. server.
4. functions.
5. roads.
6. traffic.

Examen de 1^{ère} Semestre

Questions de Cours : (06 pts)

1. Donnez la définition de l'adressage direct et l'adressage indirect
2. Donnez les étapes de cycle d'exécution d'une instruction (sans explication).
3. Donnez le rôle de décodeur, séquenceur et horloge dans le processeur.

Exercice 01 : (06 pts)

On considère un processeur à cinq étages nécessite pour accomplir l'exécution d'une instruction machine les étapes de déroulement suivants : LE, DE, CH, EX et ENR (cf. formalise vu en cours). On suppose que chaque étape prend 2 cycles d'horloge pour s'exécuter.

1. addi \$1, \$2, 10
 2. multi \$3, \$1, 20
 3. multi \$4, \$5, 2
 4. add \$6, \$5, 10
- a. Quel serait le nombre de cycles nécessaires pour exécuter ces 4 instructions sans pipeline.
 - b. Quel serait le nombre de cycles nécessaires pour exécuter ces 4 instructions avec un pipeline. Faire un schéma justifiant votre solution.
 - c. Est ce qu'on peut rencontrer des aléas lors de l'exécution de ces instructions. Quel type d'aléas ?
 - d. Proposez deux sortes d'ordonnements différents pour remédier à ce problème.

Exercice 02 : (05 pts)

Ecrire un programme en assembleur mips qui calcule La factorielle d'un nombre donnez par l'utilisateur.

Exercice 03 : (03 pts)

Donnez la représenter en code hexadécimal l'instruction suivante: ADDI \$2, \$1, -10

CORRECTION DE EXAM

Questions de Cours :

1. les (03) phases de cycle d'exécution d'une instruction sont :
 - a. Recherche de l'instruction
 - b. Décodage de l'instruction et recherche de l'opérande
 - c. Exécution de l'instruction

2. Les exemples pour chaque mode d'adressage sont les suivants :
 - a) Indirect
`ADD $s0, $s1, ($s2)`
 - b) Immédiat
`ADD $s0, $s1, 123`
 - c) registre:
`ADD $s0, $s1, $s2`

Exercice :

A : Les représentations en hexadécimal pour les instructions suivantes:

1. ADD \$s2, \$s1, \$s4

000000-00001-00100-00010-00000-100000

00000000 00000100 00000000 00000000

0024120

2. SUB \$S3, \$S7, \$S1

000000-001111-000011-000111-000000-100010

0000000000 0000000001 0000000001 0000000000 0010

00E11822

3. ADDI \$S5, \$S1, 4

001000-001011-000011-00000000000000100

0000000000 1010000000 0000000000 0000000000 0000

20A10004

4. SUB \$S2, \$S7, \$S3

000000-001111-000111-000101-000000-100010

0000000000 0000000001 0000000001 0000000000 0010

00E31022

B : Les instructions représentées par les codes hexadécimaux:

1. 00221826

000000-000011-000101-000111-000000-100110

0000000000 0000000001 0001000001 0000000000 0010

XOR \$S3,\$S1,\$S2

2. 00221824

000000-000011-000101-000111-000000-100100

0000000000 0000000001 0001000001 0000000000 0000

AND \$S3,\$S1,\$S2

B : Quelles instructions sont représentées par les codes hexadécimaux suivantes:

1. 00221827

000000 0001000011000000

NOR \$3, \$1, \$2

codop=000000=0 Donc Type R

2. 1022FFFD

000100 000101111111111101

BEQ \$1, \$2, FFFD

codop=000100 = 4 Donc Type I

CODE OP

Bits 31:29

	Bits 28:26							
	000	001	010	011	100	101	110	111
000	SPECIAL	BCOND	J	JAL	BEQ	BNE	BLEZ	BGTZ
001	ADDI	ADDIU	SLTI	SLTIU	ANDI	ORI	XORI	LUI
010	COPRO							
011								
100	LB	LH		LW	LEU	LHU		
101	SB	SH		SW				
110								
111								

SPECIAL

Bits 5:3

	Bits 2:0							
	000	001	010	011	100	101	110	111
000	SLL		SRL	SRA	SLIV		SRLV	SRAV
001	JR	JALR			SYSCALL	BREAK		
010	MFHI	MTHI	MFLO	MTLO				
011	MULT	MULTU	DIV	DIVU				
100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
101			SLT	SLTU				
110								
111								

Examen de semestre
 10/01/2017

 Documents non autorisés ; Durée  01h30.

Exercice 01 : Suivi des clients négligents (10 pts)

Une société travaille avec plus de 150 clients en comptes. En début de mois, le service "Comptabilité – Finances" demande un état des clients négligents (clients ayant plus de 5 jours de retard dans le paiement d'une facture) au service "Comptabilité-clients". Pour chacun de ces clients, une demande d'information est transmise au service commercial par le biais de la messagerie interne. Sur la base des informations recueillies, le service prend contact avec les clients pour leur faire part du retard et obtenir une explication. Si la situation financière du client ne lui apparaît pas trop critique, elle met en place de nouvelles conditions de paiement. Dans le cas contraire elle transmet le dossier au service contentieux qui démarrera une procédure judiciaire en recouvrement de créance.

1. Etablir le diagramme des flux. (5 Pts)
2. Etablir le modèle conceptuel de traitement MCT. (5 Pts)

Exercice 02 : Gestion d'une auto-école (5 pts)

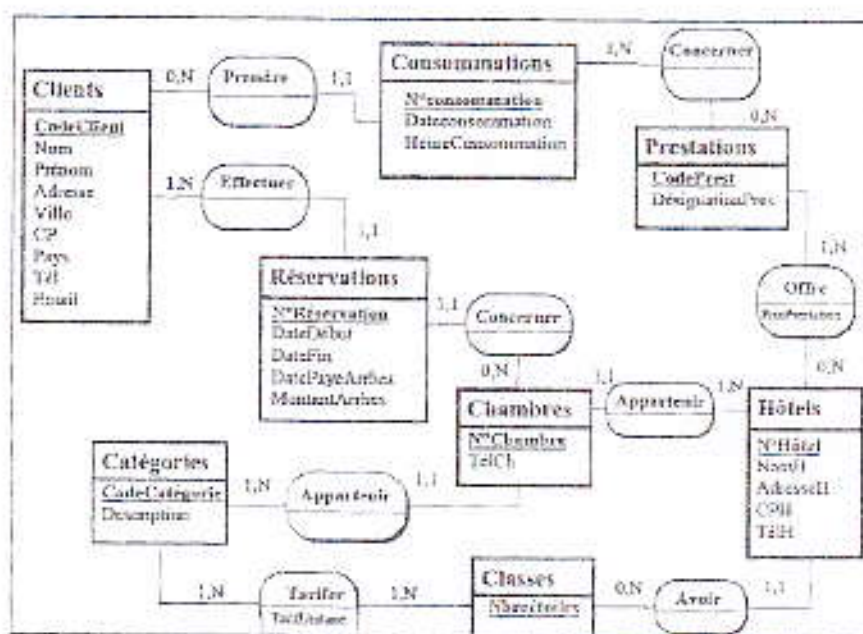
Une auto-école comprend des moniteurs et des véhicules. Elle prépare des élèves à passer un permis de type donné. Les moniteurs donnent des leçons aux élèves ; une leçon possède une durée (durelec). Les règles de gestion sont les suivantes :

- Une leçon n'est donnée que par un seul moniteur
- Un élève ne passe qu'un seul type de permis
- Une leçon est donnée que pour un seul élève
- Un élève n'apprend à conduire que sur un seul véhicule

Travail demandé : Etablir le modèle conceptuel des données correspondant.

Exercice 03 : (5 pts)

Soit le MCD suivant :



Travail demandé :
 En appliquant les règles de passage au modèle relationnel, établir le Modèle logique des données.

