

Informatique ב'

2 unités d'étude (complément pour 5 unités d'étude)

Instructions au candidat

א. Durée de l'examen : trois heures.

ב. Structure du questionnaire et barème de notation :

Ce questionnaire comporte deux parties.

Première partie — cette partie comporte quatre questions, répondez à deux d'entre elles.

(2x25) – 50 points

Deuxième partie — cette partie comporte des questions réparties en quatre sections, répondez seulement aux questions de la section dans laquelle vous avez étudié, selon les instructions indiquées dans cette section.

(2x25) – 50 points

Total – 100 points

ג. Matériel auxiliaire autorisé :

Tout matériel auxiliaire, à l'exception d'un ordinateur programmable.

ד. Instructions particulières :

(1) Tous les programmes qu'il vous est demandé d'écrire en langage informatique dans la première partie doivent être écrits dans un seul langage – C# ou Java.

(2) Inscrivez sur la couverture de votre cahier d'examen dans quel **langage** vous écrivez – C# ou Java.

(3) Inscrivez sur la couverture de votre cahier le nom de la **section** dans laquelle vous avez étudié, une des quatre sections : מערכות מחשב ואסמבלר, מערכות מחשב ואסמבלר, מודלים חישוביים, מבוא לחקר ביצועים.

Remarque : dans les programmes que vous écrivez, vous ne perdrez pas de points si vous écrivez une majuscule au lieu d'une minuscule ou l'inverse.

Tout ce que vous voulez écrire en tant que brouillon (plans, calculs, etc...), doit être écrit dans votre cahier d'examen uniquement, sur des pages séparées. Indiquez "טייטה" en haut de chaque page de brouillon.

Ecrire des notes de brouillon sur d'autres feuilles que celles du cahier d'examen, risque de vous faire disqualifier !

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:

בשאלון זה שני פרקים.

פרק ראשון - בפרק זה ארבע שאלות, ומהן יש לענות על שתיים.

(25x2) – 50 נקודות

פרק שני - בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלות רק במסלול שלמדת, לפי ההוראות בקבוצת השאלות במסלול זה.

(25x2) – 50 נקודות

סה"כ – 100 נקודות

ג. חומר עזר מותר בשימוש:

כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון כתוב בשפה אחת בלבד – C# או Java.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – C# או Java.

(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת, אחד מארבעת המסלולים: מערכות מחשב ואסמבלר, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב בטייטה (ראשי פרקים, חישובים וכדומה). רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

Bonne chance !

/Suite à la page suivante/

בהצלחה !

/המשך מעבר לדף/

Questions

Ce questionnaire comporte deux parties : une première partie et une deuxième partie. Répondez à des questions des deux parties, en suivant les instructions énoncées dans chaque partie.

Première partie (50 points)

Attention : Dans chaque question nécessitant une implémentation, vous pouvez utiliser les méthodes des classes `Node<T>`, `Queue<T>`, `Stack<T>`, `BinTreeNode<T>` sans les implémenter. Si vous utilisez des méthodes supplémentaires, implémentez-les.

Répondez à deux des questions 1-4 (chaque question vaut 25 points).

1. **Attention** : Cette question comporte deux versions : l'une en Java (pages 2-3), et l'autre en C# (pages 4-5). Composez dans le langage que vous avez appris.

Pour les élèves résolvant en Java

- ⌘. Voici plusieurs classes. Faites la trace de la méthode `main` dans la classe `Program`, puis donnez la sortie. La trace devra comporter les valeurs des variables, et les valeurs des attributs de chaque objet.

```
public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double getN() { return this.n; }
    public void f() { System.out.println("f of one "); }
    public void g() { System.out.println("g of one " + this.getN()); }
}

public class Two
{
    private Queue<One> q;
    public Two(int n, int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.insert(new One(Math.pow(2, i)));
    }
    public void f()
    {
        System.out.print("f of two ");
        if (!this.q.isEmpty())
            System.out.println(this.q.remove().getN());
    }
}
```

```
public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }

    public void methodA()
    {
        System.out.println("-- MethodA() -- ");
        this.first.f();
    }

    public void methodB()
    {
        System.out.println("-- MethodB() -- ");
        this.first.g();
        this.second.f();
    }
}

public class Program
{
    public static void main(String[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.methodA() ;
        tg.methodB() ;
    }
}
```

2. En supposant que la méthode constructeur dans la classe Together reçoit uniquement des nombres supérieurs à 0 , qu'exécute la méthode f() dans la classe Two ?

Pour les élèves résolvant en C#

- א. Voici plusieurs classes. Faites la trace de la méthode Main dans la classe Program , puis donnez la sortie. La trace devra comporter les valeurs des variables, et les valeurs des attributs de chaque objet.

```
public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double GetN() { return this.n; }
    public void F()      { Console.WriteLine("F of one "); }
    public void G()      { Console.WriteLine("G of one " + this.GetN()); }
}

public class Two
{
    private Queue<One> q;
    public Two(int n , int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.Insert(new One(Math.Pow(2 , i)));
    }
    public void F()
    {
        Console.Write("F of two ");
        if (!this.q.IsEmpty())
            Console.WriteLine(this.q.Remove().GetN());
    }
}
```

```
public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }

    public void MethodA()
    {
        Console.WriteLine("-- MethodA() -- ");
        this.first.F();
    }

    public void MethodB()
    {
        Console.WriteLine("-- MethodB() -- ");
        this.first.G();
        this.second.F();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.MethodA();
        tg.MethodB();
    }
}
```

2. En supposant que la méthode constructeur dans la classe Together reçoit uniquement des nombres supérieurs à 0 , qu'exécute la méthode F() dans la classe Two ?

2. **Attention** : Cette question comporte deux versions : l'une en Java (pages 6-7), et l'autre en C# (pages 8-9). Composez dans le langage que vous avez appris.

Pour les élèves résolvant en Java

Voici les méthodes `sod` et `what` recevant un tableau `a` trié dans l'ordre croissant dont les éléments sont de type entier, et un nombre entier `k`.

Les deux méthodes ont le même argument de sortie.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length-1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

On donne un tableau a :

2	4	7	12	18
---	---	---	----	----

- א. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `sod` pour le tableau a donné et le nombre $k = 11$. Donnez la valeur renvoyée.
 Le tableau de trace devra comporter les colonnes suivantes :
 i , j , $a[i]$, $a[j]$ et une colonne supplémentaire dans laquelle vous indiquerez si la condition de l'instruction `if` est réalisée ou non.
- ב. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `sod` pour le tableau a donné et le nombre $k = 10$. Donnez la valeur renvoyée.
 Le tableau de trace devra comporter les colonnes énumérées en א.
- ג. Quel est l'argument de sortie de la méthode `sod` ?
- ד. Quelle est la complexité du temps d'exécution de la méthode `sod` ?
 Justifiez votre réponse.
- ה. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `what` pour le tableau a donné et le nombre $k = 11$. Donnez la valeur renvoyée.
 Le tableau de trace devra comporter les colonnes suivantes :
 $left$, $right$, $a[left]$, $a[right]$ et deux colonnes supplémentaires pour chacune des instructions `if`. Dans chaque colonne, vous indiquerez si la condition de l'instruction `if` est réalisée ou non.
- ו. Quelle est la complexité du temps d'exécution de la méthode `what` ?
 Justifiez votre réponse.
- ז. Parmi les deux méthodes `sod` et `what`, laquelle est la plus efficace ?
 Justifiez votre réponse.
- ח. L'argument d'entrée des méthodes `sod` et `what` a été modifié de telle sorte qu'il est possible de leur transmettre un tableau a non trié.
 - (1) L'argument de sortie de la méthode `sod` sera-t-il modifié ?
 Justifiez votre réponse.
 - (2) L'argument de sortie de la méthode `what` sera-t-il modifié ?
 Justifiez votre réponse.

Pour les élèves résolvant en C#

Voici les méthodes Sod et What recevant un tableau a trié dans l'ordre croissant dont les éléments sont de type entier, et un nombre entier k .

Les deux méthodes ont le même argument de sortie.

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length-1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

On donne un tableau a : a

2	4	7	12	18
---	---	---	----	----

א. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `Sod` pour le tableau a donné et le nombre $k = 11$. Donnez la valeur renvoyée.

Le tableau de trace devra comporter les colonnes suivantes :

$i, j, a[i], a[j]$ et une colonne supplémentaire dans laquelle vous indiquerez si la condition de l'instruction `if` est réalisée ou non.

ב. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `Sod` pour le tableau a donné et le nombre $k = 10$. Donnez la valeur renvoyée.

Le tableau de trace devra comporter les colonnes énumérées en א.

ג. Quel est l'argument de sortie de la méthode `Sod` ?

ד. Quelle est la complexité du temps d'exécution de la méthode `Sod` ? Justifiez votre réponse.

ה. A l'aide d'un tableau de trace, tracez l'exécution de la méthode `What` pour le tableau a donné et le nombre $k = 11$. Donnez la valeur renvoyée.

Le tableau de trace devra comporter les colonnes suivantes :

`left, right, a[left], a[right]` et deux colonnes supplémentaires pour chacune des instructions `if`. Dans chaque colonne, vous indiquerez si la condition de l'instruction `if` est réalisée ou non.

ו. Quelle est la complexité du temps d'exécution de la méthode `What` ? Justifiez votre réponse.

ז. Parmi les deux méthodes `Sod` et `What`, laquelle est la plus efficace ? Justifiez votre réponse.

ח. L'argument d'entrée des méthodes `Sod` et `What` a été modifié de telle sorte qu'il est possible de leur transmettre un tableau a non trié.

(1) L'argument de sortie de la méthode `Sod` sera-t-il modifié ? Justifiez votre réponse.

(2) L'argument de sortie de la méthode `What` sera-t-il modifié ? Justifiez votre réponse.

3. **Attention** : Cette question comporte deux versions : l'une en Java (pages 10-11), et l'autre en C# (pages 12-13). Composez dans le langage que vous avez appris.

Pour les élèves résolvant en Java

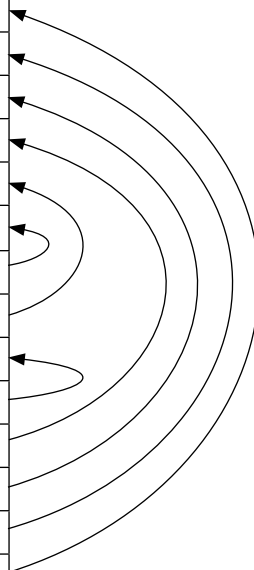
Voici une interface de la classe "תור-ביטול" (UndoQueue) :

La classe définit un type collection avec un protocole FIFO insérant et retirant des valeurs entières et supérieures à 0 .

Nom de la méthode	Description
UndoQueue()	La méthode construit un תור-ביטול vide.
boolean isEmpty()	La méthode renvoie true si le תור-ביטול présent est vide, autrement, la méthode renvoie false .
void insert(int x)	La méthode insère la valeur x à la fin du תור-ביטול présent.
int remove()	La méthode retire la valeur se trouvant au début du תור-ביטול présent, et la renvoie. <u>Supposition</u> : le תור-ביטול présent n'est pas vide.
int head()	La méthode renvoie la valeur se trouvant au début du תור-ביטול sans la retirer. <u>Supposition</u> : le תור-ביטול présent n'est pas vide.
void undo()	La méthode annule la dernière méthode remove ou la dernière méthode insert exécutée, et rétablit le תור-ביטול tel qu'il était avant l'exécution de la méthode. Si aucune méthode remove ou insert n'a été exécutée, la méthode n'effectue aucune action. <u>Attention</u> : l'exécution n fois de la méthode undo , rétablira le תור-ביטול dans l'état où il était avant l'exécution des n dernières méthodes remove ou insert qui n'ont pas encore été annulées.

Exemple :

Etat du תור-ביטול après l'exécution de la méthode	La méthode
Le תור-ביטול est vide	UndoQueue uq = new UndoQueue();
→2→	uq.insert(2);
→3 , 2→	uq.insert(3);
→3→	uq.remove();
→4 , 3→	uq.insert(4);
→4→	uq.remove();
Le תור-ביטול est vide	uq.remove();
→4→	uq.undo();
→4 , 3→	uq.undo();
→1 , 4 , 3→	uq.insert(1);
→4 , 3→	uq.undo();
→3→	uq.undo();
→3 , 2→	uq.undo();
→2→	uq.undo();
Le תור-ביטול est vide	uq.undo();
Le תור-ביטול est vide	uq.undo();



א. Voici une séquence de code :

```
UndoQueue q = new UndoQueue();
q.insert(1);
q.insert(2);
q.insert(3);
q.remove();
q.insert(4);
q.undo();
q.undo();
```

Donnez l'état du **תור-ביטול** à la suite de l'exécution de chacune des instructions de la séquence de code.

- ב. Ecrivez en Java la déclaration de la classe **UndoQueue** et ses attributs. Ecrivez une documentation pour chaque attribut.
- ג. Implémentez en Java les méthodes insert , remove et undo figurant dans l'interface de la classe **UndoQueue** .

Vous pouvez utiliser les autres méthodes de l'interface de **UndoQueue** sans les implémenter. Si vous utilisez des méthodes supplémentaires, implémentez-les.

Pour les élèves résolvant en C#

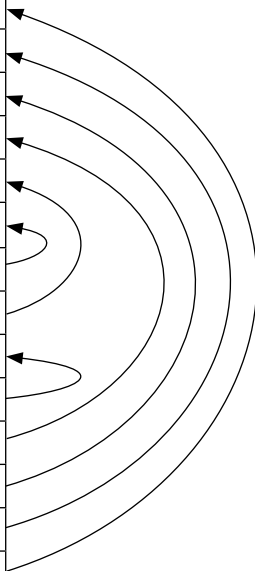
Voici une interface de la classe "תור-ביטול" (UndoQueue) :

La classe définit un type collection avec un protocole FIFO insérant et retirant des valeurs entières et supérieures à 0 .

Nom de la méthode	Description
UndoQueue()	La méthode construit un תור-ביטול vide.
bool IsEmpty()	La méthode renvoie true si le תור-ביטול présent est vide, autrement, la méthode renvoie false .
void Insert(int x)	La méthode insère la valeur x à la fin du תור-ביטול présent.
int Remove()	La méthode retire la valeur se trouvant au début du תור-ביטול présent, et la renvoie. <u>Supposition</u> : le תור-ביטול présent n'est pas vide.
int Head()	La méthode renvoie la valeur se trouvant au début du תור-ביטול sans la retirer. <u>Supposition</u> : le תור-ביטול présent n'est pas vide.
void Undo()	La méthode annule la dernière méthode Remove ou la dernière méthode Insert exécutée, et rétablit le תור-ביטול tel qu'il était avant l'exécution de la méthode. Si aucune méthode Remove ou Insert n'a été exécutée, la méthode n'effectue aucune action. <u>Attention</u> : l'exécution n fois de la méthode Undo , rétablira le תור-ביטול dans l'état où il était avant l'exécution des n dernières méthodes Remove ou Insert qui n'ont pas encore été annulées.

Exemple :

Etat du תור-ביטול après l'exécution de la méthode	La méthode
Le תור-ביטול est vide	UndoQueue uq = new UndoQueue();
→2→	uq.Insert(2);
→3 , 2→	uq.Insert(3);
→3→	uq.Remove();
→4 , 3→	uq.Insert(4);
→4→	uq.Remove();
Le תור-ביטול est vide	uq.Remove();
→4→	uq.Undo();
→4 , 3→	uq.Undo();
→1 , 4 , 3→	uq.Insert(1);
→4 , 3→	uq.Undo();
→3→	uq.Undo();
→3 , 2→	uq.Undo();
→2→	uq.Undo();
Le תור-ביטול est vide	uq.Undo();
Le תור-ביטול est vide	uq.Undo();



א. Voici une séquence de code :

```
UndoQueue q = new UndoQueue();
q.Insert(1);
q.Insert(2);
q.Insert(3);
q.Remove();
q.Insert(4);
q.Undo();
q.Undo();
```

Donnez l'état du תור-ביטול à la suite de l'exécution de chacune des instructions de la séquence de code.

- ב. Ecrivez en C# la déclaration de la classe **UndoQueue** et ses attributs. Ecrivez une documentation pour chaque attribut.
- ג. Implémentez en C# les méthodes **Insert** , **Remove** et **Undo** figurant dans l'interface de la classe **UndoQueue** .

Vous pouvez utiliser les autres méthodes de l'interface de **UndoQueue** sans les implémenter. Si vous utilisez des méthodes supplémentaires, implémentez-les.

/suite en page 14/

4. א. Voici la déclaration d'une méthode :

En Java :

```
public static void leaves(BinTreeNode<Integer> t , Stack<Integer> s)
```

En C# :

```
public static void Leaves(BinTreeNode<int> t , Stack<int> s)
```

Cette méthode reçoit un arbre binaire t de nombres entiers non vide, et une pile s de nombres entiers vide.

La méthode insère dans la pile les valeurs des feuilles de l'arbre t , selon un ordre de recherche de droite à gauche.

Implémentez cette méthode en Java ou en C# .

- ב. Ecrivez en Java ou en C#, une méthode booléenne recevant 2 arbres binaires de nombres entiers non vides, et qui renverra `true` si ces deux conditions sont réalisées :

- ils ont le même nombre de feuilles
- d'après l'ordre de recherche de droite à gauche, les valeurs des feuilles sont égales

Autrement, la méthode renverra `false` .

Utilisez les méthodes que vous avez implémentées en א .

Deuxième partie (50 points)

Cette partie comporte des questions concernant quatre sections différentes :

מערכות מחשב ואסמבלר, pages 15-20

מבוא לחקר ביצועים, pages 21-31

מודלים חישוביים, pages 32-34

תכנות מונחה עצמים ב-C#, pages 46-55 ; תכנות מונחה עצמים ב-Java, pages 36-45

Répondez uniquement aux questions de la section dans laquelle vous avez étudié.

מערכות מחשב ואסמבלר

Si vous avez étudié dans cette section, répondez à deux des questions 5-8 (25 points par section).

5. Dans une pile sont stockés 10 nombres entiers avec signe, de la taille d'un mot. De plus, on définit dans le segment de données une variable K et deux tableaux, ARR1 et ARR2 .
Toutes les cases du tableau ARR1 sont initialisées à 0 , et la variable K est initialisée à 0 .

ARR1	DW	10	DUP	(0)
ARR2	DW	10	DUP	(?)
K	DW	0		

On doit copier dans le tableau ARR1 tous les nombres stockés dans la pile de la manière suivante :

Tous les nombres négatifs seront stockés au début du tableau selon leur ordre de sortie de la pile. Après les nombres négatifs, on stockera tous les zéros, puis tous les nombres positifs seront stockés d'après leur ordre de sortie de la pile.

Par exemple, pour la pile ci-dessous contenant 6 éléments :

-5
0
10
-2
0
8

le tableau ARR1 paraîtra ainsi :

0	1	2	3	4	5
-5	-2	0	0	10	8

Voici une séquence de programme en assembleur, effectuant cela.

Des lignes ont été effacées de la séquence de programme aux emplacements signalés par les numéros i-iv.

Notez dans votre cahier les numéros des lignes marquées, et écrivez à côté de chacun des numéros la ligne d'instruction manquante, afin que la séquence de programme effectue ce qui est demandé.

		MOV	SI , 0
		MOV	DI , 0
		MOV	CX , 10
	A1:	POP	AX
i		_____	_____
		JLE	A2
ii		_____	_____
		ADD	DI , 2
		JMP	A4
	A2:	CMP	AX , 0
		JNE	A3
		INC	K
		JMP	A4
iii	A3:	_____	_____
		ADD	SI , 2
	A4:	LOOP	A1
		ADD	SI , K
		ADD	SI , K
		MOV	CX , DI
		SHR	CX , 1
		XOR	BX , BX
	A5:	MOV	DX , ARR2[BX]
		MOV	ARR1[SI] , DX
iv		_____	_____
		ADD	BX , 2
		LOOP	A5

6. Un tableau A est désigné comme **contenu** dans un tableau B, si la longueur du tableau A n'est pas supérieure à la longueur du tableau B, et si toutes les valeurs des éléments du tableau A se trouvent dans le tableau B d'affilée, et dans le même ordre que dans le tableau A.

Dans chaque tableau sont stockés des nombres entiers avec signe, différents entre eux.

Exemple :

Concernant les tableaux A et B ci-dessous, le tableau A est **contenu** dans le tableau B.

Tableau A :

0	1	2	3	4	5
3	-5	0	8	5	11

Tableau B :

0	1	2	3	4	5	6	7	8	9
15	-7	4	3	-5	0	8	5	11	2

Les éléments du tableau B signalés en gris sont les valeurs des éléments du tableau A.

⌘. Les données ont été ainsi définies dans le segment de données :

ARR_B DB 100 DUP (?)

ARR_A DB 10 DUP (?)

V DB ?

P DB ?

Ecrivez en assembleur une routine (procédure) nommée TEST, qui recevra une valeur stockée dans la variable V. La routine vérifiera si la valeur stockée dans la variable V se trouve dans le tableau ARR_B. Si tel est le cas, la routine stockera dans la variable P l'indice de l'élément correspondant, autrement, la routine stockera la valeur -1 dans la variable P.

2. Ecrivez en assembleur une séquence de programme qui vérifiera si le tableau ARR_A est **contenu** dans le tableau ARR_B. Si c'est le cas, la séquence de programme stockera 1 dans le registre BL, autrement, elle stockera 0 dans le registre BL.

Utilisez la routine TEST que vous avez écrite en ⌘.

Supposez que la longueur du tableau ARR_A n'est pas supérieure à la longueur du tableau ARR_B.

/suite en page 18/

7. Cette question comporte deux paragraphes, א et ב , qui ne sont pas liés.
 Répondez aux deux paragraphes.

א. Un tableau ARR est défini dans le segment de données :

ARR DB 6 DUP (?)

On veut inverser l'ordre des éléments dans le tableau.

On donne par exemple ce tableau :

0	1	2	3	4	5
3	2	10	0	6	7

A la suite de l'inversion de l'ordre des éléments, le tableau paraîtra ainsi :

0	1	2	3	4	5
7	6	0	10	2	3

Voici 2 séquences en assembleur, i et ii , dont le but est d'inverser l'ordre des éléments dans le tableau ARR .

A l'aide d'un tableau de trace, tracez l'exécution de chacune des séquences i et ii suivantes, puis déterminez si elle effectue ce qui est demandé ou non.

i

```

MOV SI, 3
MOV DI, 2
A1: MOV AL, ARR[DI]
     MOV AH, ARR[SI]
     ADD ARR[DI], AH
     SUB ARR[DI], AL
     MOV ARR[SI], AL
     INC SI
     DEC DI
     JNZ A1
    
```

ii

```

MOV CX, 3
MOV SI, 3
MOV DI, 2
A1: MOV AL, ARR[DI]
     MOV AH, ARR[SI]
     MOV ARR[DI], AH
     MOV ARR[SI], AL
     INC SI
     DEC DI
     LOOP A1
    
```

- ב. (Il n'y a pas de lien avec le paragraphe א.)

Les données ont été définies dans le segment de données de la manière suivante :

T DB 55 , 90 , 110 , 1

Attention : les nombres sont décimaux.

Voici une séquence de programme en assembleur, dont le but est de calculer la somme des éléments du tableau T .

	MOV	CX , 4
	XOR	AX , AX
	LEA	BX , T
AGAIN:	ADD	AL , [BX]
	INC	BX
	LOOP	AGAIN

La séquence de programme n'effectue pas ce qui est demandé.

- (1) A l'aide d'un tableau de trace, tracez l'exécution de la séquence de programme, puis donnez le contenu de AL et le contenu de l'indicateur de retenue et de l'indicateur de débordement au terme de l'exécution de la séquence de programme.
- (2) Modifiez la séquence de programme de façon à ce que soit calculée la somme des éléments du tableau T donné.

Copiez dans votre cahier la séquence de programme après la modification.

8. Voici une séquence de programme en assembleur :

- i PUSHF
- ii PUSH BP
- iii MOV BP , SP
- iv MOV AX , [BP+4]
- v MOV BP , [BP]
- vi ADD SP , 4
- vii POPF

On donne qu'avant l'exécution de la séquence de programme :

- le contenu du registre des indicateurs est 3202H
- le contenu du registre SP est 96H
- le contenu du registre BP est 5678H

Voici une partie d'une pile contenant des nombres hexadécimaux.

Avant l'exécution de la séquence de programme, le registre SP pointe sur la case indiquée par une flèche, dont l'adresse relative est 96H .

							↓				
.....	*	*	*	*	*	*	02H	32H	*	*	
décalage des cases	90H	91H	92H	93H	94H	95H	96H	97H	98H	99H	

Lorsqu'une case est marquée par une * , son contenu n'a pas d'importance.

א. A l'aide d'un tableau de trace, tracez l'exécution de la séquence de programme.

Le tableau de trace devra inclure des colonnes pour les registres AX , BP et SP .

ב. Répondez aux paragraphes (1)-(4) en vous basant sur le tableau de trace.

- (1) Quel nombre s'inscrira dans la pile donnée à la suite de l'exécution de l'instruction i ? Donnez l'adresse relative des cases dans lesquelles le nombre s'inscrira.
- (2) Quel sera le contenu du registre BP à la suite de l'exécution de l'instruction iii ?
- (3) Quel sera le contenu du registre AX à la suite de l'exécution de l'instruction iv ?
- (4) Quel sera le contenu du registre BP à la suite de l'exécution de l'instruction v ?

מבוא לחקר ביצועים

Si vous avez étudié dans cette section, répondez à deux des questions 9-12 (25 points par question).

9. Cette question comporte trois paragraphes, א-ג, qui ne sont pas dépendants l'un de l'autre. Répondez à tous les paragraphes.

א. Soit le graphe orienté $G = (V, E)$, V étant l'ensemble des sommets et E étant l'ensemble des arcs.

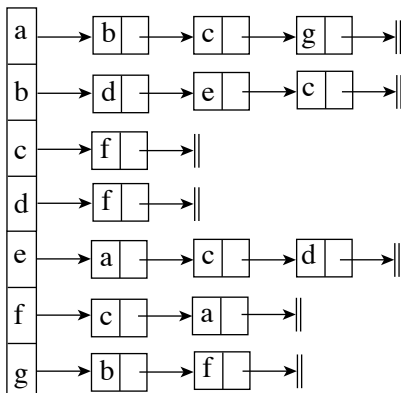
G est représenté par la matrice d'adjacences suivante :

	a	b	c	d	e
a	0	1	1	0	0
b	0	0	1	0	0
c	0	1	0	1	0
d	0	0	0	0	1
e	1	0	1	0	0

- (1) Tracez le graphe G représenté par la matrice.
- (2) Trouvez la / les composante(s) fortement connexe(s) (CFC) du graphe G . Pour chaque CFC trouvée, donnez l'ensemble de ses sommets.
- (3) Retirez un arc du graphe G que vous avez tracé, de sorte que le nombre de CFC soit incrémenté de 1. Indiquez quel arc vous avez retiré, ainsi que l'ensemble des sommets de chaque CFC.

ב. Soit le graphe orienté $G = (V, E)$, V étant l'ensemble des sommets et E étant l'ensemble des arcs.

G est représenté par la liste d'adjacences suivante :



(1) Appliquez un algorithme de recherche en profondeur (DFS) sur le graphe donné en commençant par le sommet a .

Tracez dans votre cahier l'arbre couvrant DFS obtenu.

(2) Appliquez un algorithme de recherche en largeur (BFS) sur le graphe donné en commençant par le sommet d .

Tracez dans votre cahier l'arbre couvrant BFS obtenu.

ג. Soit un graphe orienté G .

On applique l'algorithme de recherche en profondeur (DFS) sur le graphe orienté G .

On définit les arcs du graphe G à l'aide de l'arbre couvrant / dans la forêt couvrante obtenu(e).

קשת עץ est un arc du graphe G , se trouvant également dans l'arbre couvrant / dans la forêt couvrante de G .

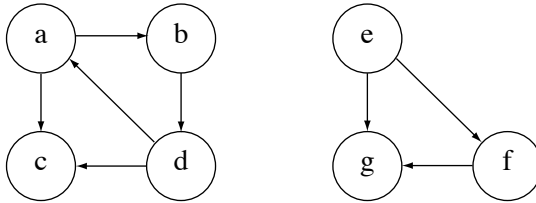
קשת אחורית est un arc (u, v) du graphe G , qui ne se trouve pas dans l'arbre couvrant / dans la forêt couvrante. C'est un arc du graphe G allant du sommet u au sommet v , v étant un ancêtre de u dans l'arbre couvrant / dans la forêt couvrante.

קשת קדמית est un arc (u, v) du graphe G qui ne se trouve pas dans l'arbre couvrant / dans la forêt couvrante. C'est un arc du graphe G allant du sommet u au sommet v , v étant un descendant de u dans l'arbre couvrant / dans la forêt couvrante.

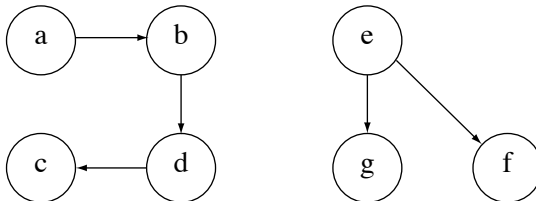
קשת חוצה est un arc (u, v) du graphe G , qui ne se trouve pas dans l'arbre couvrant / dans la forêt couvrante, et qui n'est

ni **קשת אחורית** ni **קשת קדמית**. C'est un arc du graphe G allant du sommet u au sommet v , à la condition qu'aucun de ces deux sommets ne soit le descendant de l'autre dans l'arbre couvrant / dans la forêt couvrante.

- (1) Voici un graphe orienté G , comportant plus d'une CFC (composante fortement connexe).



A la suite de l'application de l'algorithme de recherche en profondeur (DFS) sur le graphe orienté G donné, on obtient la forêt couvrante suivante :



Copiez le graphe G sur votre cahier. Basez-vous sur la forêt couvrante donnée et notez à côté de chacun des arcs du graphe G s'il est :

- קשת עץ
 - קשת אחורית
 - קשת קדמית
 - קשת חוצה
- (2) Voici une affirmation dans laquelle il manque une expression à l'emplacement signalé par un trait :

Si après l'application d'un algorithme de recherche en profondeur (DFS) sur un graphe orienté G , on trouve dans G _____, alors G comporte un cycle / des cycles.

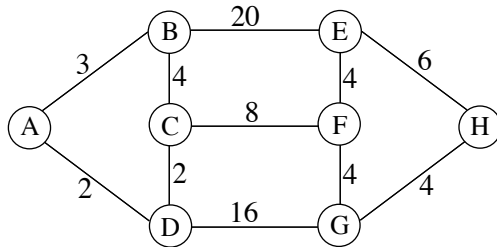
Voici quatre propositions pour compléter l'affirmation :

- קשת עץ
- קשת אחורית
- קשת קדמית
- קשת חוצה

Copiez dans votre cahier la proposition qui complète correctement l'affirmation, puis justifiez votre réponse.

10. Cette question comporte deux paragraphes, א et ב , qui ne sont pas dépendants l'un de l'autre. Répondez aux deux paragraphes.

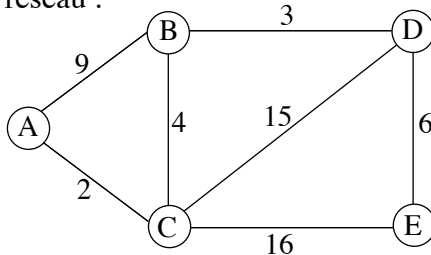
א. Voici un réseau $G = (V, E)$, V étant l'ensemble des sommets et E étant l'ensemble des arcs :



Trouvez l'arbre couvrant minimal en utilisant l'algorithme de Prim, puis tracez l'arbre couvrant minimal obtenu pour le réseau donné, en commençant du sommet C .

ב. (Il n'y a pas de lien avec le paragraphe א .)

On donne le réseau :



- (1) Trouvez l'arbre couvrant minimal en utilisant l'algorithme de Kruskal, puis tracez l'arbre couvrant minimal obtenu pour le réseau donné.
- (2) Un élève a proposé à sa professeur un algorithme permettant de trouver l'arbre couvrant minimal d'un réseau $G = (V, E)$, V étant l'ensemble des sommets et E étant l'ensemble des arcs du graphe. Voici l'algorithme de l'élève :

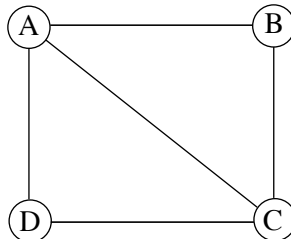
Etape 1 : Créer un graphe G_1 , dont l'ensemble des sommets est l'ensemble des sommets du réseau G et dont l'ensemble des arcs E_T est un ensemble vide.

- Etape 2 : Trier les arcs du réseau G en ordre croissant, selon leurs poids respectifs. L'ensemble des arcs triés obtenu est noté E^* .
- Etape 3 : Pour tout arc (u, v) dans E^* , en commençant par l'arc ayant le poids le plus petit, exécuter :
si l'ajout de l'arc (u, v) ne crée pas de cycle dans G_1 , ajouter l'arc (u, v) à E_T .
autrement, renvoyer le graphe G_1 qui est un arbre couvrant minimal.
- Etape 4 : Renvoyer le graphe G_1 qui est un arbre couvrant minimal.

Appliquez l'algorithme de l'élève sur le réseau donné.

Répondez aux paragraphes (i)-(iv).

- (i) Tracez dans votre cahier G_1 qui s'est formé à la suite de l'exécution de l'étape 1.
 - (ii) Inscrivez dans votre cahier tous les arcs de E^* à la suite de l'exécution de l'étape 2.
 - (iii) Tracez dans votre cahier chacun des graphes G_1 obtenus au cours de l'étape 3.
 - (iv) Tracez dans votre cahier l'arbre couvrant renvoyé par l'algorithme.
- (3) La professeur a répondu à l'élève que l'algorithme qu'il a proposé est erroné, car sa sortie n'est pas nécessairement un arbre couvrant minimal. Par exemple, pour ce réseau :



il est possible d'associer des poids aux arcs du réseau de telle sorte que si on utilise l'algorithme de l'élève, l'arbre couvrant minimal ne sera pas renvoyé.

Copiez dans votre cahier le réseau, puis inscrivez des poids à côté des arcs, de telle sorte que l'arbre renvoyé par l'algorithme de l'élève ne soit pas un arbre couvrant minimal. /suite en page 26/

11. Cette question comporte deux paragraphes, א et ב, qui ne sont pas dépendants l'un de l'autre. Répondez aux deux paragraphes.

א. On donne un problème de programmation linéaire : $\max \{z = kx_1 + 20x_2\}$
 soumis aux contraintes suivantes :

$$-3x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \geq 0$$

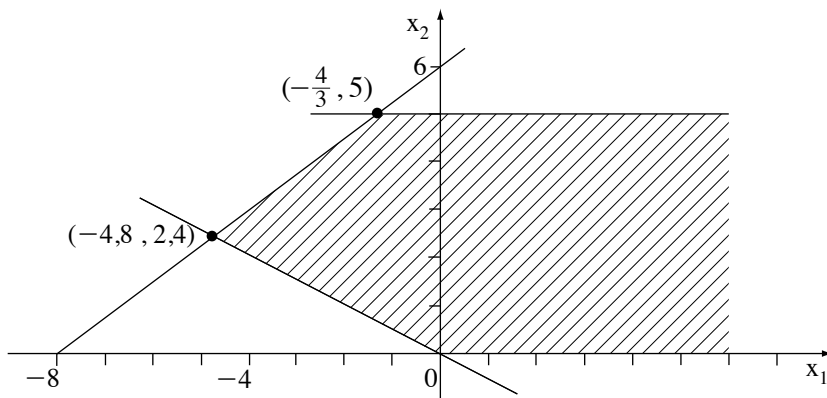
$$x_2 \leq 5$$

$$x_2 \geq 0$$

x_1 est sans restriction de signe.

k est un paramètre.

Voici le tracé du domaine des solutions admissibles au problème donné.



Dans chacun des quatre paragraphes (1)-(4) suivants, une certaine valeur du paramètre k est donnée.

(1) $k = 20$.

(2) $k = -20$.

(3) $k = -15$.

(4) $k = -15$, en ajoutant la contrainte : $x_1 \leq -5$.

Les paragraphes ne sont pas dépendants l'un de l'autre.

Pour chacun des quatre paragraphes (1)-(4) :

- Si la solution optimale est unique, trouvez cette solution optimale unique, ainsi que la valeur de la fonction objectif de cette solution.
- S'il existe une infinité de solutions optimales, donnez la solution optimale générale au problème, ainsi que la valeur de la fonction objectif de cette solution.
- Si la solution optimale n'est pas bornée, justifiez votre réponse.
- S'il n'y a pas de solution admissible, justifiez votre réponse.

/suite en page 27/

ב. (Il n'y a pas de lien avec le paragraphe α.)

On donne un problème de programmation linéaire :

$$\max \{z = ax_1 + 3x_2\}$$

soumis aux contraintes suivantes :

$$x_1 + \frac{3}{8}x_2 \leq 3$$

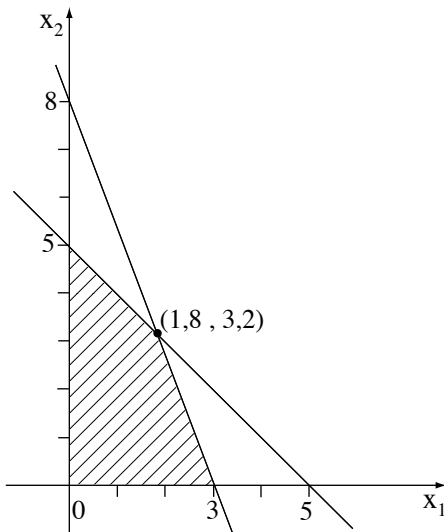
$$x_1 + x_2 \leq 5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$a > 0$ est un paramètre du problème.

Voici le tracé du domaine des solutions admissibles au problème donné.



Voici quatre domaines, i-iv , concernant le paramètre a :

i $a < 3$

ii $a > 8$

iii $3 < a < 8$

iv $3 < a < 24$

Déterminez pour lequel des domaines i-iv , la solution $(1,8 , 3,2)$ sera une solution optimale **unique**. Copiez le domaine dans votre cahier, puis justifiez votre réponse.

12. Cette question comporte cinq paragraphes, א-ה , qui ne dépendent pas l'un de l'autre. Répondez à tous les paragraphes.

א. Dans le tableau ci-dessous, on donne une partie d'une solution de base admissible au problème de transport :

$$x_{11} = 80, x_{21} = 20.$$

Origines	Destinations			Offre
	1	2	3	
1	10 80	15	17	80
2	10 20	18	14	120
3	15	20	18	50
Demande	100	100	50	

Copiez le tableau dans votre cahier, puis complétez-le selon la méthode du coin nord-ouest.

א. Dans le tableau suivant, on donne une partie d'une solution de base admissible à un problème de transport, et on donne des valeurs de $v_1, v_2, v_3, u_1, u_2, u_3$.

Origines	Destinations			Offre	u_i
	1	2	3		
1	10 80	15	12	80	0
2	10	100	10	100	3
3	20	10	50	50	-5
Demande	80	100	50		
v_j	10	15	19		

Copiez le tableau dans votre cahier, puis complétez-le selon la méthode du coin nord-ouest en tenant compte des valeurs des u_i et des v_j , de manière à obtenir une solution de base admissible.

/suite en page 29/

- ג. Dans le tableau suivant, on donne une solution de base admissible à un problème de transport, et on donne des valeurs de $v_1, v_2, v_3, u_1, u_2, u_3$.

Origines	Destinations			Offre	u_i
	1	2	3		
1	10 100	18	13 30	130	0
2	6 100	21	12	100	-4
3	11	16 50	14 100	150	1
Demande	200	50	130		
v_j	10	15	13		

La solution est-elle optimale ? Justifiez votre réponse.

7. Dans le tableau suivant, on donne une solution optimale à un problème de transport, et on donne les valeurs de $v_1, v_2, v_3, u_1, u_2, u_3$ correspondant à cette solution.

Origines	Destinations			Offre	u_i
	1	2	3		
1	10	25	16	20	10
	20				
2	10	22	14	50	10
	30		20		
3	18	20	20	60	16
		40	20		
Demande	50	40	40		
v_j	0	4	4		

- (1) La solution donnée est-elle optimale unique ? Justifiez votre réponse.
 (2) Dans le tableau donné ci-dessus, soustrayez 4 du coût à l'unité de chaque origine vers chaque destination. Tracez le tableau ainsi obtenu.

Déterminez laquelle des quatre affirmations i-iv suivantes est correcte.

Copiez-la dans votre cahier, puis justifiez votre réponse.

- i La solution donnée n'est pas une solution admissible au problème obtenu.
 ii La solution donnée est une solution de base admissible mais non optimale au problème obtenu.
 iii La solution donnée est une solution optimale unique au problème obtenu.
 iv La solution donnée est une solution optimale mais n'est pas une solution optimale unique au problème obtenu.

- ה. Dans le tableau suivant, on donne une solution non optimale obtenue au terme de k itérations, à un problème de transport donné, et on donne des valeurs de $v_1, v_2, v_3, v_4, u_1, u_2, u_3$.

Origines	Destinations				Offre	u_i
	1	2	3	4		
1	5 (150)	2 (80)	4 -4	2 -6	230	0
2	2 (70)	3 4	5 (80)	3 -2	150	-3
3	7 7	4 7	3 (200)	3 (340)	540	-5
Demande	220	80	280	340		$Z = 3070$
v_j	5	2	8	8		

La solution décrite dans ce tableau n'est pas optimale du fait de la présence de variables hors base dont la valeur du $(c_{ij} - u_i - v_j)$ est négative.

Effectuez une itération supplémentaire, c'est à dire une itération $k + 1$.

- (1) Quelle variable sort de la base lors de cette itération ?
- (2) Tracez dans votre cahier un nouveau tableau, et inscrivez-y la solution obtenue à la suite de cette itération.

מודלים חישוביים

Si vous avez étudié dans cette section, répondez à deux des questions 13-16 (25 points par question).

13. Voici le langage L dans l'alphabet $\{a, b, c\}$:

$$L = \{a^n b^{3k+1} c^k \mid n > 0, k > 0\}$$

א. Ecrivez le mot le plus court du langage L .

ב. Construisez un automate pile acceptant le langage L .

14. Cette question comporte deux paragraphes, א et ב , qui ne sont pas liés.
Répondez aux deux paragraphes.

א. Voici le langage L dans l'alphabet $\{a, b\}$:

$$L = \{w \mid |w| > 0, \text{ le dernier caractère de } w \text{ apparaît dans l'ensemble du mot un nombre pair de fois} \}$$

Exemple :

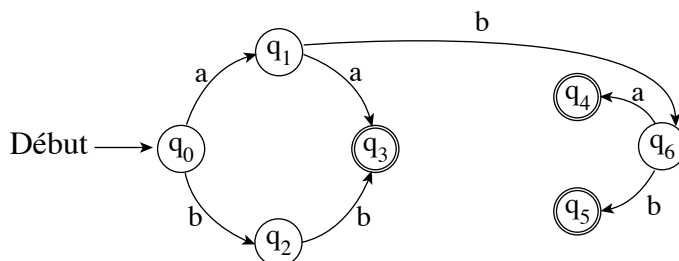
Le mot abaabbab appartient au langage.

Le mot abbabaa appartient au langage.

Le mot babbaba n'appartient pas au langage.

Voici le diagramme partiel d'un automate fini déterministe acceptant le langage L .

Il manque des transitions et des symboles d'entrée à ce diagramme.



Le diagramme comporte tous les états de l'automate, ainsi que tous les états acceptants.

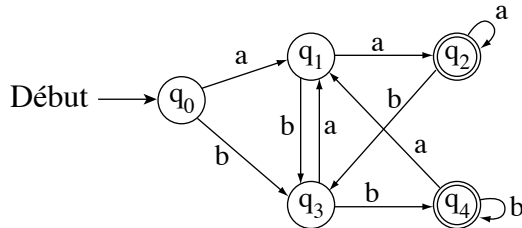
Copiez ce diagramme dans votre cahier, puis complétez-le de façon à ce que l'automate accepte le langage L .

Complétez les transitions manquantes et les symboles d'entrée manquants.

Attention : il ne faut ni ajouter ni retirer d'états à l'automate.

- ב. (Il n'y a pas de lien avec le paragraphe א.)

Voici un automate fini déterministe acceptant le langage L dans l'alphabet $\{a, b\}$:



- (1) Pour chacun des quatre mots (i)-(iv) ci-dessous, déterminez s'il est accepté par l'automate. Si le mot est accepté par l'automate, donnez le chemin acceptant pour ce mot.

- (i) aaba
- (ii) bbaabb
- (iii) abaa
- (iv) bb

- (2) Quel est le langage L défini par l'automate ?

15. Voici le langage L dans l'alphabet $\{0, 1, 2\}$:

$$L = \{0^n 1^k 2 \mid n > k \geq 0\}$$

- א. Démontrez que le langage L n'est pas régulier.

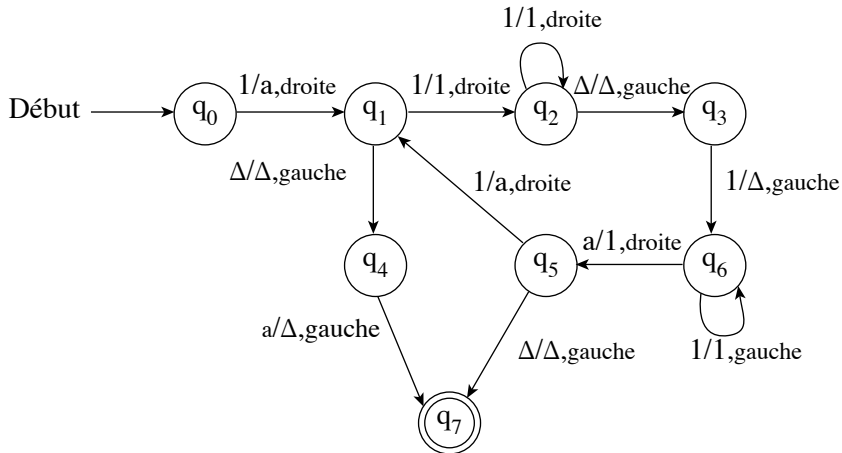
- ב. On donne le langage L_1 dans l'alphabet $\{1, 2\}$:

$$L_1 = \{1^n 2 \mid n \geq 0\}$$

Quel est le langage $L_1 \cap L$? Justifiez.

16. Voici une machine de Turing calculant la fonction $f(x)$.

La machine reçoit en entrée un nombre x entier supérieur à 0, inscrit sur le ruban en nombre unaire par x caractères 1, qui sont suivis par le symbole Δ . La machine inscrit le résultat du calcul de $f(x)$ sur le ruban en nombre unaire après le symbole $\vdash$, et à la suite de ce nombre apparaît le symbole Δ .



א. Quel sera le contenu du ruban après le calcul de $f(3)$? Montrez le processus de calcul de la machine.

Pour chaque étape, donnez l'état du ruban, indiquez où se trouve la tête de lecture et notez dans lequel des états q_0 - q_7 se trouve la machine.

ב. Quel sera le contenu du ruban après le calcul de $f(5)$?

ג. Quel sera le contenu du ruban après le calcul de $f(6)$?

ד. Quelle est la fonction $f(x)$ que la machine calcule ?

ה. Ajoutez à la machine une transition / des transitions afin qu'elle calcule également $f(0)$.

Notez dans votre cahier la transition / les transitions que vous avez ajoutée(s).

Pour chaque transition, donnez : de quel état elle provient, vers quel état elle est dirigée, ainsi que ce qui y est inscrit.

Attention : la programmation orientée objet débute
en page 36

תכנות מונחה עצמים

Si vous avez étudié dans cette section et que vous rédigez en Java, répondez à deux des questions 17-20 (25 points par question).

17. Voici les classes **AA** et **BB** :

```
public class AA
{
    private String st;

    public AA()                { this.st = "excellent"; }
    public AA(String st)       { this.st = st; }
    public String getSt()      { return this.st; }
    public void setSt(String st) { this.st = st; }
    public String toString()   { return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB()                { super(); this.num = 1; }
    public BB(int num , String st) { super(st); this.num = Math.abs(num); }
    public int getNum()         { return this.num; }
    public void setNum(int num) { this.num = num; }
    public String toString()    { return super.toString() + " num = " + this.num; }
}
```

- א. Définissez dans la classe **AA** une méthode booléenne nommée `isLike(Object obj)` recevant un objet `obj` de type `Object` . Si l'objet `obj` est de type **AA** et que le contenu de la chaîne `st` de `obj` est identique au contenu de la chaîne `st` de l'objet en présence, la méthode renverra `true` , autrement, elle renverra `false` .
- ב. Définissez dans la classe **BB** une méthode redéfinissant la méthode que vous avez définie en א .
Si l'objet `obj` est de type **BB** et que la valeur de son attribut `num` est identique à la valeur de l'attribut `num` de l'objet en présence, la méthode renverra `true` , autrement, elle renverra `false` .

- ג. Voici un extrait issu de la méthode principale :

```
AA a = new AA("excellent");  
BB b = new BB();  
a = b;  
if (a.isLike(b)) System.out.println(a);
```

La séquence de programme est-elle valide ?

Si tel est le cas, quelle sera la sortie de cette séquence ? Indiquez quelle version de la méthode `isLike` s'exécutera : celle de `AA` ou celle de `BB` . Sinon, expliquez quelle est l'erreur, et à quel moment elle se révélera : au moment de la compilation ou au moment de l'exécution.

- ד. Voici un extrait issu d'une méthode principale :

```
AA aa = new AA();  
BB bb = new BB(2 , "excellent");  
bb = aa;  
if (bb.isLike(aa)) System.out.println(bb);
```

La séquence de programme est-elle valide ?

Si c'est le cas, quelle sera la sortie de cette séquence ? Indiquez quelle version de la méthode `isLike` s'exécutera : celle de `AA` ou celle de `BB` . Sinon, expliquez quelle est l'erreur, et à quel moment elle se révélera : au moment de la compilation ou au moment de l'exécution.

- ה. Ecrivez une méthode externe nommée `longString` recevant un tableau d'objets de type `Object` . La méthode renvoie une chaîne constituée de l'enchaînement de l'attribut `st` des objets de type `AA` du tableau, de la manière suivante :

- Si l'objet n'a que `st` comme attribut, la chaîne de l'attribut `st` sera enchaînée une fois.
- Si l'objet a également `num` comme attribut, la chaîne de l'attribut `st` sera enchaînée un nombre `num` de fois.
- Si aucun objet de type `AA` n'est présent dans le tableau, une chaîne vide sera renvoyée.

18. Voici un projet comportant les classes : **B** , **D** , **A** et **OopTest** :

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        System.out.println("B(" + m1 + ", " + m2 + "), #" + numB);
    }
}

public class D extends B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x)
    {
        super(x, x);
        this.d = d;
        numD++;
        System.out.println("D(" + d + ", " + x + "), #" + numD);
    }

    public D(double d, int x, int y)
    {
        super(x , y);
        this.d = d;
        numD++;
        System.out.println("D(" + d + ", " + x + ", " + y + "), #" + numD);
    }
}
```

/suite en page 39/

```
public class A
{
    private static int numA = 0;
    private A a;
    private B b;

    public A(A a , B b)
    {
        this.a = a;
        this.b = b;
        numA++;
        System.out.println("A Constructor , #" + numA);
    }
}

public class OopTest
{
    public static void main(String [] args)
    {
        B w1 = new B(2, 3);
        B w2 = new D(1.5 , 6);
        B w3 = new D(2.3 , 8 , 9);
        A w4 = new A(null , w1);
        A w5 = new A(w4 , w3);
    }
}
```

Ecrivez la trace de la méthode main dans la classe **OopTest** , et donnez la sortie.

La trace devra comporter les valeurs des variables, ainsi que les valeurs des attributs de chaque objet.

19. Une société éditant des logiciels éducatifs a développé un logiciel traitant de suites de nombres entiers.

Le système a été développé par étapes.

Pour chaque suite de nombres, on considère :

- (1) Le premier terme de la suite de rang 1 .
- (2) Le terme de rang n dans la suite.
- (3) L'affichage des n premiers termes de la suite.

Lors de la première étape, deux classes ont été développées :

Une suite arithmétique (**ASeq**) – une suite dans laquelle la différence entre chaque terme et celui qui le précède est une valeur constante.

Une suite géométrique (**GSeq**) – une suite dans laquelle le quotient entre chaque terme et celui qui le précède est une valeur constante.

Voici le code des classes développées lors de la première étape :

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int theNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

```
public void displayNElements(int n)
{
    System.out.print("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        System.out.print(this.theNElement(i+1) + " , ");
    System.out.println(this.theNElement(n));
}
}
```

```
public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first , int product)
    {
        this.first = first;
        this.product = product;
    }

    public int theNElement(int n)
    {
        return this.first * (int)Math.pow(this.product , n - 1);
    }

    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0 ; i < n - 1; i++)
            System.out.print(this.theNElement(i+1) + " , ");
        System.out.println(this.theNElement(n));
    }
}
```

- א. Tracez la séquence de programme suivante. Représentez-y l'objet construit, ses attributs et la sortie.

```
ASeq aSeq = new ASeq(2 , 3);  
System.out.println(aSeq.theNElement(4));  
aSeq.displayNElements(5);
```

Lors de la deuxième étape du développement, il a été décidé qu'il convenait de développer une nouvelle classe décrivant une suite constante (**Sequence**), de telle sorte que les classes **ASeq** et **GSeq** héritent de cette nouvelle classe. Dans une suite constante, la valeur du premier terme est définie, et tous les autres termes sont identiques au premier terme.

- ב. Complétez le développement de la deuxième étape de la façon correspondant le mieux aux principes de la programmation orientée objet, et selon les instructions (i)-(ii) :
- (i) Implémentez entièrement la classe mère **Sequence** . Cette classe doit prendre en considération :
 - (1) Le premier terme de la suite de rang 1.
 - (2) Le terme de rang n dans la suite.
 - (3) L'affichage des n premiers termes de la suite.
 - (ii) Implémentez à nouveau la classe **ASeq** de sorte qu'elle hérite de la classe **Sequence** .

Lors de la troisième étape du développement, il a été décidé d'étendre le projet incluant les trois classes développées lors de la deuxième étape (**GSeq** , **ASeq** , **Sequence**), de sorte qu'il soit possible d'exécuter une méthode calculant et renvoyant la somme des n premiers termes de chaque suite. Supposez que la classe **GSeq** a été de nouveau implémentée, de sorte qu'elle hérite de la classe **Sequence** .

- ג. Pour chacune des classes **GSeq** , **ASeq** , **Sequence** , indiquez s'il est nécessaire de lui apporter des modifications telles que le projet réponde aux exigences de développement de la troisième étape , de la façon correspondant le mieux aux principes de la programmation orientée objet.
- Si des modifications doivent être apportées, détaillez et implémentez-les.

Lors de la quatrième étape du développement, il a été décidé de développer une méthode statique `check` recevant une valeur entière n , ainsi que deux objets de suites : l'un de type **ASeq** et l'autre de type **GSeq**.

La méthode calcule la somme des n premiers termes de chacune des deux suites puis renvoie :

- Le caractère 'A', si la somme des n premiers termes de la suite de type **ASeq** est la plus grande des deux sommes.
 - Le caractère 'G', si la somme des n premiers termes de la suite de type **GSeq** est la plus grande des deux sommes.
 - Le caractère 'E', si les sommes des n premiers termes des deux suites sont égales.
7. Implémentez la méthode statique `check` d'après les exigences définies lors de la quatrième étape du développement.

20. Une clinique vétérinaire destinée aux animaux de compagnie, regroupe des informations sur les vétérinaires exerçant dans la clinique et sur les animaux de compagnie qui y sont soignés. Le nombre de vétérinaires exerçant dans cette clinique est de 10 au maximum, et le nombre d'animaux de compagnie qui y sont soignés est de 500 au maximum.

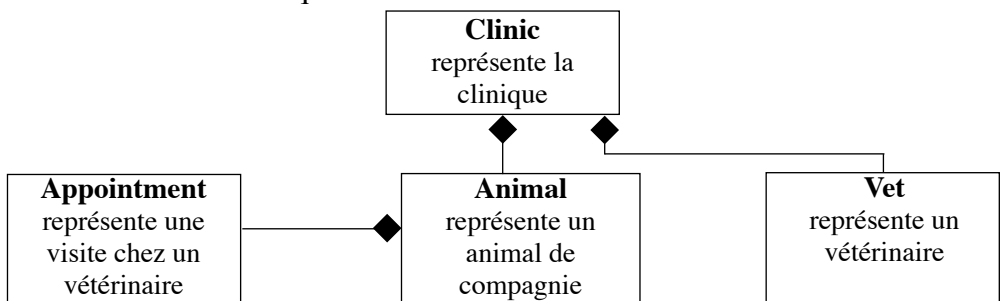
Les lignes directrices de la gestion des données sont :

- Pour chaque vétérinaire, sont enregistrés : le numéro d'identité, le nom, l'ancienneté en années.
- Pour chaque animal de compagnie, sont enregistrés : le numéro d'immatriculation, le nom, le type (par exemple : chien, chat ou lapin), l'âge, le détail des 50 (au maximum) dernières visites à la clinique, le nombre de visites réservées.
- Pour chaque visite à la clinique, sont enregistrés : le numéro d'identité du vétérinaire ayant soigné l'animal de compagnie lors de cette visite, ainsi qu'une chaîne de caractères contenant les codes des soins que l'animal de compagnie a reçus lors de cette même visite. Le code d'un soin est une lettre capitale parmi toutes les lettres de l'alphabet.

Parmi les opérations que la méthode peut effectuer, il y a :

- La production d'un compte-rendu de tous les animaux de compagnie d'un type particulier (par exemple, tous les lapins). Pour chaque animal, le compte-rendu inclura son numéro d'immatriculation, son nom, et son âge.
- Au début de chaque année : une mise à jour de l'âge de l'animal de compagnie, c'est-à-dire un incrément de 1 de l'âge, et une mise à jour de l'ancienneté de chaque vétérinaire, c'est-à-dire un incrément de 1 de l'ancienneté.
- Le renvoi du nom d'un vétérinaire d'après son numéro d'identité.

Voici le diagramme de la hiérarchie des classes exigées pour la gestion des données de cette clinique.



Dans le diagramme, le symbole —◆— représente une inclusion.

/suite en page 45/

- א. Définissez les attributs et les méthodes de chaque classe du diagramme. Définissez uniquement les méthodes nécessaires afin de répondre aux exigences décrites en début de question (en page 44) et dans le diagramme.

Supposez que sont données des méthodes constructeur, recevant un paramètre pour chaque attribut, et des méthodes accesseur (méthodes `get`), et qu'il n'est pas nécessaire de les écrire. Les méthodes mutateur (méthodes `set`) ne sont pas données.

Pour chaque attribut, écrivez en Java sa définition, et écrivez sa documentation.

Pour chaque méthode, écrivez en Java sa déclaration, et écrivez une documentation incluant ce qu'elle reçoit et ce qu'elle renvoie. Il n'est pas nécessaire d'implémenter la méthode.

- ב. Supposez que le système a été développé conformément au plan que vous avez présenté en א, et que toutes les méthodes données et celles que vous avez définies sont implémentées.

Dans la classe **Clinic**, on ajoute la méthode :

```
public void addAppointment (Animal p, String t, Vet v)
```

recevant un animal de compagnie `p`, une chaîne `t` de codes de soins que l'animal a reçus lors de la visite actuelle, et le vétérinaire `v` qui l'a soigné. La méthode ajoute la visite à l'animal de compagnie.

Implémentez entièrement cette méthode.

Supposez que le nombre des visites précédentes de cet animal à la clinique est inférieur à 50.

Si en plus des méthodes données et des méthodes que vous avez définies en א, vous utilisez d'autres méthodes, implémentez-les entièrement et indiquez pour chacune des méthodes dans quelle classe il faut l'implémenter.

תכנות מונחה עצמים

Si vous avez étudié dans cette section et que vous rédigez en C#, répondez à deux des questions 21-24 (25 points par question).

21. Voici les classes **AA** et **BB** :

```
public class AA
{
    private string st;

    public AA() { this.st = "excellent"; }
    public AA(string st) { this.st = st; }
    public string GetSt() { return this.st; }
    public void SetSt (string st) { this.st = st; }
    public override string ToString() { return "st = " + this.st; }
}

public class BB : AA
{
    private int num;

    public BB() : base() { this.num = 1; }
    public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }
    public int GetNum() { return this.num; }
    public void SetNum(int num) { this.num = num; }
    public override string ToString() { return base.ToString() + " num = "
                                     + this.num; }
}
```

- א. Définissez dans la classe **AA** une méthode booléenne de redéfinition nommée `IsLike(Object obj)` recevant un objet `obj` de type `Object`. Si l'objet `obj` est de type **AA** et que le contenu de la chaîne `st` de `obj` est identique au contenu de la chaîne `st` de l'objet en présence, la méthode renverra `true`, autrement, elle renverra `false`.
- ב. Définissez dans la classe **BB** une méthode redéfinissant la méthode que vous avez définie en א.

Si l'objet `obj` est de type **BB** et que la valeur de son attribut `num` est identique à la valeur de l'attribut `num` de l'objet en présence, la méthode renverra `true`, autrement, elle renverra `false`.

/suite en page 47/

- ג. Voici un extrait issu de la méthode principale :

```
AA a = new AA("excellent");  
BB b = new BB();  
a = b;  
if (a.IsLike(b)) Console.WriteLine(a);
```

La séquence de programme est-elle valide ?

Si tel est le cas, quelle sera la sortie de cette séquence ? Indiquez quelle version de la méthode `IsLike` s'exécutera : celle de `AA` ou celle de `BB` . Sinon, expliquez quelle est l'erreur, et à quel moment elle se révélera : au moment de la compilation ou au moment de l'exécution.

- ד. Voici un extrait issu d'une méthode principale :

```
AA aa = new AA();  
BB bb = new BB(2 , "excellent");  
bb = aa;  
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

La séquence de programme est-elle valide ?

Si c'est le cas, quelle sera la sortie de cette séquence ? Indiquez quelle version de la méthode `IsLike` s'exécutera : celle de `AA` ou celle de `BB` . Sinon, expliquez quelle est l'erreur, et à quel moment elle se révélera : au moment de la compilation ou au moment de l'exécution.

- ה. Ecrivez une méthode externe nommée `LongString` recevant un tableau d'objets de type `Object` . La méthode renvoie une chaîne constituée de l'enchaînement de l'attribut `st` des objets de type `AA` du tableau, de la manière suivante :

- Si l'objet n'a que `st` comme attribut, la chaîne de l'attribut `st` sera enchaînée une fois.
- Si l'objet a également `num` comme attribut, la chaîne de l'attribut `st` sera enchaînée un nombre `num` de fois.
- Si aucun objet de type `AA` n'est présent dans le tableau, une chaîne vide sera renvoyée.

22. Voici un projet comportant les classes : **B** , **D** , **A** et **OopTest** :

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        Console.WriteLine("B(" + m1 + ", " + m2 + ") , #" + numB);
    }
}

public class D : B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x) : base (x , x)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ") , #" + numD);
    }

    public D(double d, int x, int y) : base(x , y)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ", " + y + ") , #" + numD);
    }
}
```

```
public class A
{
    private static int numA = 0;
    private A a;
    private B b;

    public A(A a , B b)
    {
        this.a = a;
        this.b = b;
        numA++;
        Console.WriteLine("A Constructor , #" + numA);
    }
}
```

```
public class OopTest
{
    public static void Main(string[] args)
    {
        B w1 = new B(2 , 3);
        B w2 = new D(1.5 , 6);
        B w3 = new D(2.3 , 8 , 9);
        A w4 = new A(null , w1);
        A w5 = new A(w4 , w3);
    }
}
```

Ecrivez la trace de la méthode Main dans la classe **OopTest** , et donnez la sortie.

La trace devra comporter les valeurs des variables, ainsi que les valeurs des attributs de chaque objet.

23. Une société éditant des logiciels éducatifs a développé un logiciel traitant de suites de nombres entiers.

Le système a été développé par étapes.

Pour chaque suite de nombres, on considère :

- (1) Le premier terme de la suite de rang 1 .
- (2) Le terme de rang n dans la suite.
- (3) L'affichage des n premiers termes de la suite.

Lors de la première étape, deux classes ont été développées :

Une suite arithmétique (**ASeq**) – une suite dans laquelle la différence entre chaque terme et celui qui le précède est une valeur constante.

Une suite géométrique (**GSeq**) – une suite dans laquelle le quotient entre chaque terme et celui qui le précède est une valeur constante.

Voici le code des classes développées lors de la première étape :

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int TheNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

```
public void DisplayNElements(int n)
{
    Console.Write("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        Console.Write(this.TheNElement(i+1) + " , ");
    Console.WriteLine(this.TheNElement(n));
}
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }

    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product , n - 1);
    }

    public void DisplayNElements(int n)
    {
        Console.Write("The sequence elements: ");
        for (int i = 0; i < n - 1; i++)
            Console.Write(this.TheNElement(i+1) + " , ");
        Console.WriteLine (this.TheNElement(n));
    }
}
```

(Attention : la suite de la question se trouve à la page suivante.)

/suite en page 52/

- א. Tracez la séquence de programme suivante. Représentez-y l'objet construit, ses attributs et la sortie.

```
ASeq aSeq = new ASeq(2 , 3);  
Console.WriteLine(aSeq.TheNElement(4));  
aSeq.DisplayNElements(5);
```

Lors de la deuxième étape du développement, il a été décidé qu'il convenait de développer une nouvelle classe décrivant une suite constante (**Sequence**), de telle sorte que les classes **ASeq** et **GSeq** héritent de cette nouvelle classe. Dans une suite constante, la valeur du premier terme est définie, et tous les autres termes sont identiques au premier terme.

- א. Complétez le développement de la deuxième étape de la façon correspondant le mieux aux principes de la programmation orientée objet, et selon les instructions (i)-(ii) :
- (i) Implémentez entièrement la classe mère **Sequence** . Cette classe doit prendre en considération :
 - (1) Le premier terme de la suite de rang 1.
 - (2) Le terme de rang n dans la suite.
 - (3) L'affichage des n premiers termes de la suite.
 - (ii) Implémentez à nouveau la classe **ASeq** de sorte qu'elle hérite de la classe **Sequence** .

Lors de la troisième étape du développement, il a été décidé d'étendre le projet incluant les trois classes développées lors de la deuxième étape (**GSeq** , **ASeq** , **Sequence**), de sorte qu'il soit possible d'exécuter une méthode calculant et renvoyant la somme des n premiers termes de chaque suite. Supposez que la classe **GSeq** a été de nouveau implémentée, de sorte qu'elle hérite de la classe **Sequence** .

- א. Pour chacune des classes **GSeq** , **ASeq** , **Sequence** , indiquez s'il est nécessaire de lui apporter des modifications telles que le projet réponde aux exigences de développement de la troisième étape , de la façon correspondant le mieux aux principes de la programmation orientée objet.

Si des modifications doivent être apportées, détaillez et implémentez-les.

Lors de la quatrième étape du développement, il a été décidé de développer une méthode statique `Check` recevant une valeur entière n , ainsi que deux objets de suites : l'un de type `ASeq` et l'autre de type `GSeq`. La méthode calcule la somme des n premiers termes de chacune des deux suites puis renvoie :

- Le caractère 'A', si la somme des n premiers termes de la suite de type `ASeq` est la plus grande des deux sommes.
 - Le caractère 'G', si la somme des n premiers termes de la suite de type `GSeq` est la plus grande des deux sommes.
 - Le caractère 'E', si les sommes des n premiers termes des deux suites sont égales.
7. Implémentez la méthode statique `Check` d'après les exigences définies lors de la quatrième étape du développement.

24. Une clinique vétérinaire destinée aux animaux de compagnie, regroupe des informations sur les vétérinaires exerçant dans la clinique et sur les animaux de compagnie qui y sont soignés. Le nombre de vétérinaires exerçant dans cette clinique est de 10 au maximum, et le nombre d'animaux de compagnie qui y sont soignés est de 500 au maximum.

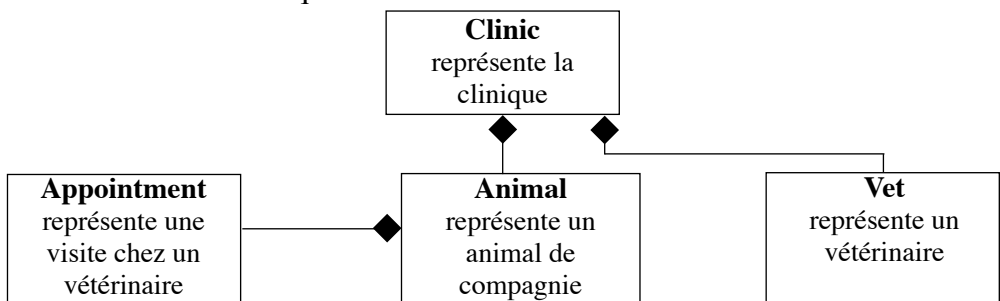
Les lignes directrices de la gestion des données sont :

- Pour chaque vétérinaire, sont enregistrés : le numéro d'identité, le nom, l'ancienneté en années.
- Pour chaque animal de compagnie, sont enregistrés : le numéro d'immatriculation, le nom, le type (par exemple : chien, chat ou lapin), l'âge, le détail des 50 (au maximum) dernières visites à la clinique, le nombre de visites réservées.
- Pour chaque visite à la clinique, sont enregistrés : le numéro d'identité du vétérinaire ayant soigné l'animal de compagnie lors de cette visite, ainsi qu'une chaîne de caractères contenant les codes des soins que l'animal de compagnie a reçus lors de cette même visite. Le code d'un soin est une lettre capitale parmi toutes les lettres de l'alphabet.

Parmi les opérations que la méthode peut effectuer, il y a :

- La production d'un compte-rendu de tous les animaux de compagnie d'un type particulier (par exemple, tous les lapins). Pour chaque animal, le compte-rendu inclura son numéro d'immatriculation, son nom, et son âge.
- Au début de chaque année : une mise à jour de l'âge de l'animal de compagnie, c'est-à-dire un incrément de 1 de l'âge, et une mise à jour de l'ancienneté de chaque vétérinaire, c'est-à-dire un incrément de 1 de l'ancienneté.
- Le renvoi du nom d'un vétérinaire d'après son numéro d'identité.

Voici le diagramme de la hiérarchie des classes exigées pour la gestion des données de cette clinique.



Dans le diagramme, le symbole —◆— représente une inclusion.

/suite en page 55/

- א. Définissez les attributs et les méthodes de chaque classe du diagramme. Définissez uniquement les méthodes nécessaires afin de répondre aux exigences décrites en début de question (en page 54) et dans le diagramme.

Supposez que sont données des méthodes constructeur, recevant un paramètre pour chaque attribut, et des méthodes accesseur (méthodes Get), et qu'il n'est pas nécessaire de les écrire. Les méthodes mutateur (méthodes Set) ne sont pas données.

Pour chaque attribut, écrivez en C# sa définition, et écrivez sa documentation.

Pour chaque méthode, écrivez en C# sa déclaration, et écrivez une documentation incluant ce qu'elle reçoit et ce qu'elle renvoie. Il n'est pas nécessaire d'implémenter la méthode.

- ב. Supposez que le système a été développé conformément au plan que vous avez présenté en א , et que toutes les méthodes données et celles que vous avez définies sont implémentées.

Dans la classe **Clinic** , on ajoute la méthode :

```
public void AddAppointment (Animal p , string t , Vet v)
```

recevant un animal de compagnie p , une chaîne t de codes de soins que l'animal a reçus lors de la visite actuelle, et le vétérinaire v qui l'a soigné. La méthode ajoute la visite à l'animal de compagnie.

Implémentez entièrement cette méthode.

Supposez que le nombre des visites précédentes de cet animal à la clinique est inférieur à 50.

Si en plus des méthodes données et des méthodes que vous avez définies en א , vous utilisez d'autres méthodes, implémentez-les entièrement et indiquez pour chacune des méthodes dans quelle classe il faut l'implémenter.

Bonne chance !

בהצלחה !