

NOM :

Juin 2023

PRÉNOM :

BLOC :



Examen de Mathématiques 2 :

1^{ère} année Bachelier en Informatique de Gestion

BINV1100 – Mathématiques 2

Date : 7 juin 2022

Durée de l'examen : 2 heures et 30 minutes

Nombre de questions : 5

- 1. Sauf avis contraire, toute réponse doit être justifiée.**
2. Si vous n'écrivez pas proprement et lisiblement, votre réponse recevra un zéro.
3. Écrire au crayon est autorisé si le point 2 ci-dessus est respecté.
4. Vous pouvez avoir à votre disposition 10 feuilles recto/verso respectant les conditions suivantes : vos nom et prénom doivent être indiqués, les feuilles doivent être manuscrites, reliées sur toute la longueur de manière à ne pas pouvoir en détacher sans l'arracher et le contenu ne fait pas l'objet de miniaturisation.
5. Pour les questions sur machine, vous devez travailler **sur le U** : . En effet, si vous travaillez ailleurs vos fichiers seront perdus.
6. Les points communiqués en regard des questions sont indicatifs. Des lacunes graves entraîneront l'échec au présent examen.
- 7. Mettez vos noms et prénoms au début de chaque question !**

Question 1	/20
Question 2	/20
Question 3	/20
Question 4	/15
Question 5	/15
TOTAL	/90

Nom :

Prénom :

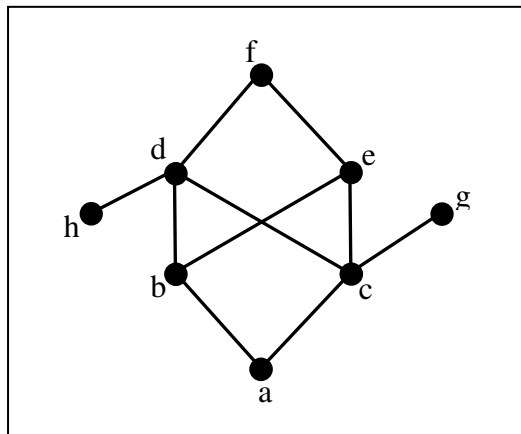
PARTIE I : SUR PAPIER

Nom :

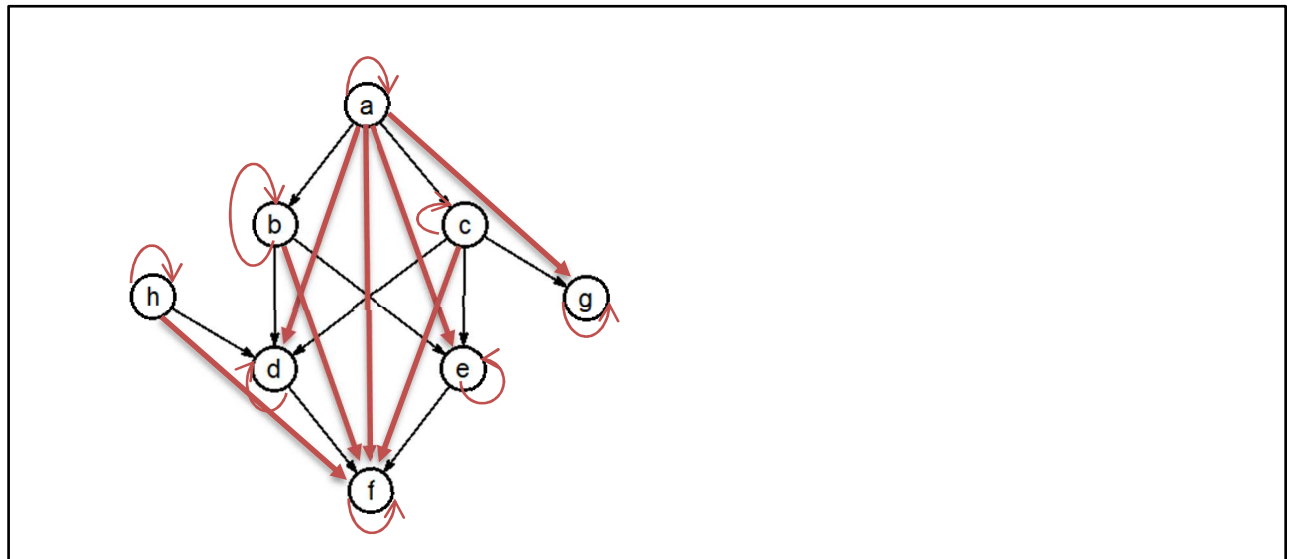
Prénom :

Question 1 (20 pts)

D Soit un ordre partiel \leq sur l'ensemble $E = \{a, b, c, d, e, f, g, h\}$, défini par le diagramme de Hasse ci-dessous :



a) Complétez le digraphe qui suit pour qu'il représente la même relation :



b) Pour chacun des ordres totaux ci-dessous, indiquez « oui » si c'est un tri topologique de \leq , et « non » sinon. Si c'est « non », justifiez votre assertion.

1 : $a \rightarrow b \rightarrow c \rightarrow g \rightarrow e \rightarrow h \rightarrow d \rightarrow f$

2 : $a \rightarrow b \rightarrow h \rightarrow c \rightarrow e \rightarrow d \rightarrow g \rightarrow f$

3 : $a \rightarrow c \rightarrow g \rightarrow e \rightarrow b \rightarrow h \rightarrow d \rightarrow f$

1 : Oui

2 : Oui

3 : Non car $b < e$. Donc b doit être pris avant e

Nom :

Prénom :

c) L'ordre partiel \leq sur E est-il un treillis ? Justifiez votre réponse.

Non car f et g sont deux maximaux non comparable donc la paire {f,g} n'a pas de supremum.

d) Indiquez si \leq sur E possède un minimum, un maximum, un infimum et un suprémum. Dans l'affirmative, donnez l'élément. Dans la négative, justifiez.

- **E n'a pas de minimum car E a deux minimaux (a et h) non comparables**
- **E n'a pas de maximum car E a deux maximaux (f et g) non comparables**
- **E n'a pas d'infimum car E a deux minimaux**
- **E n'a pas de supremum car E a deux maximaux**

e) Relativement à \leq sur E, déterminez les 4 ensembles suivants : les minimaux, les maximaux, les minorants et les majorants.

- **Minimaux = {a,h}**
- **Maximaux = {f,g}**
- **Minorant = { }**
- **Majorant = { }**

f) Proposez un sous-ensemble de E possédant au moins 5 éléments et qui soit un treillis pour \leq .

{a,b,c,d,f} car la seule paire d'élément non comparable est {b,c}

avec $\inf(\{b,c\}) = a$ et $\sup(\{b,c\}) = d$

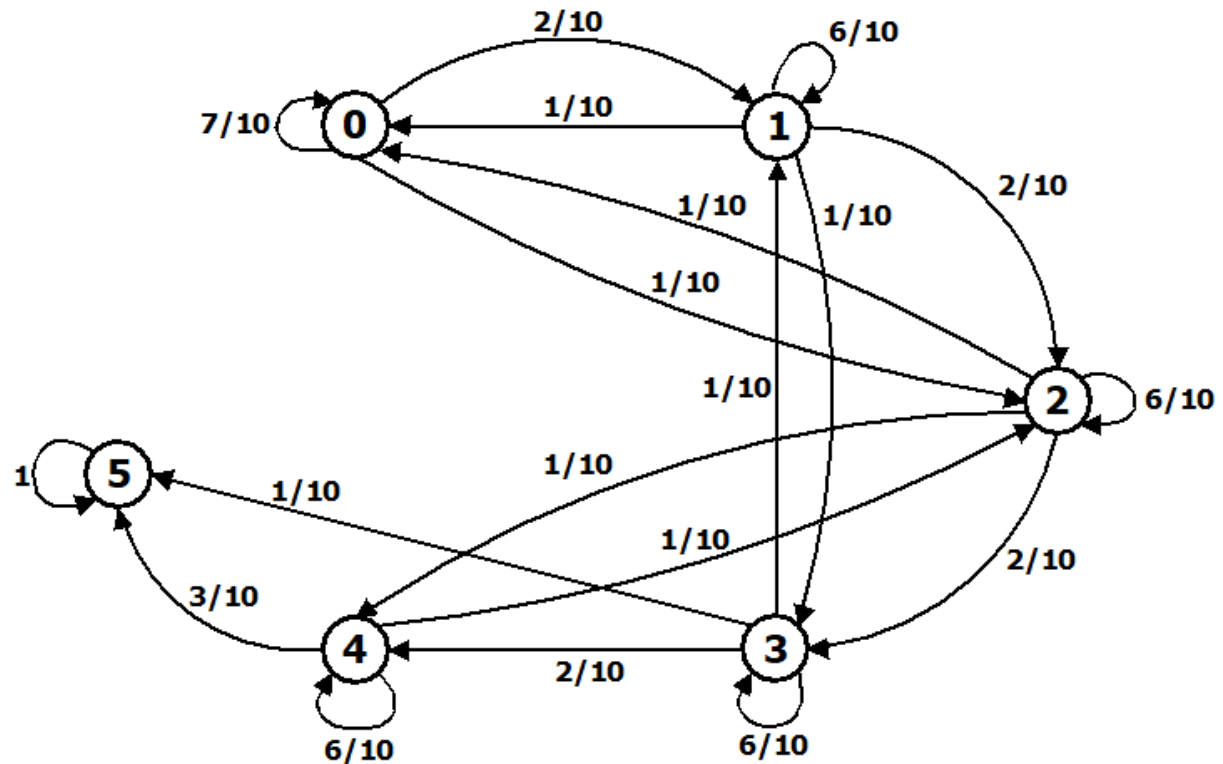
Nom :

Prénom :

Question 2 (20 pts)

- 1) Le Zamcha est un jeu traditionnel qui se joue avec un dé à dix faces. Chaque joueur lance à son tour le dé. S'il fait 8 ou plus il gagne un point. S'il fait un 10, il gagne un point supplémentaire (sans pouvoir monter au-dessus de 5). En revanche, s'il fait un 1, il perd 2 points (sans pouvoir descendre en-dessous de 0). Il faut cinq points pour gagner la partie.

a. Complétez le diagramme du processus de Markov et écrivez la matrice de transition.



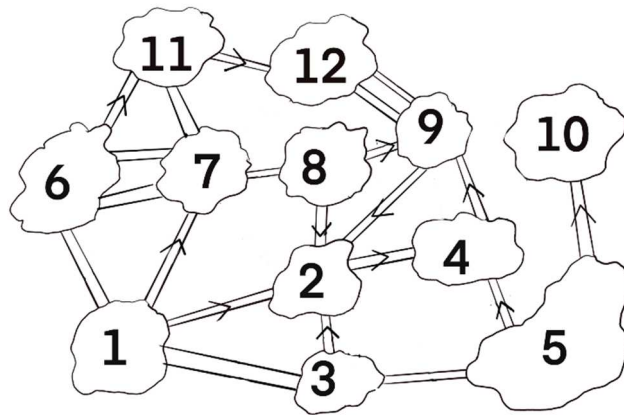
Matrice de transition :

$$P = \begin{pmatrix} 7/10 & 1/10 & 1/10 & 0 & 0 & 0 \\ 2/10 & 6/10 & 0 & 1/10 & 0 & 0 \\ 1/10 & 2/10 & 6/10 & 0 & 1/10 & 0 \\ 0 & 1/10 & 2/10 & 6/10 & 0 & 0 \\ 0 & 0 & 1/10 & 2/10 & 6/10 & 0 \\ 0 & 0 & 0 & 1/10 & 3/10 & 1 \end{pmatrix}$$

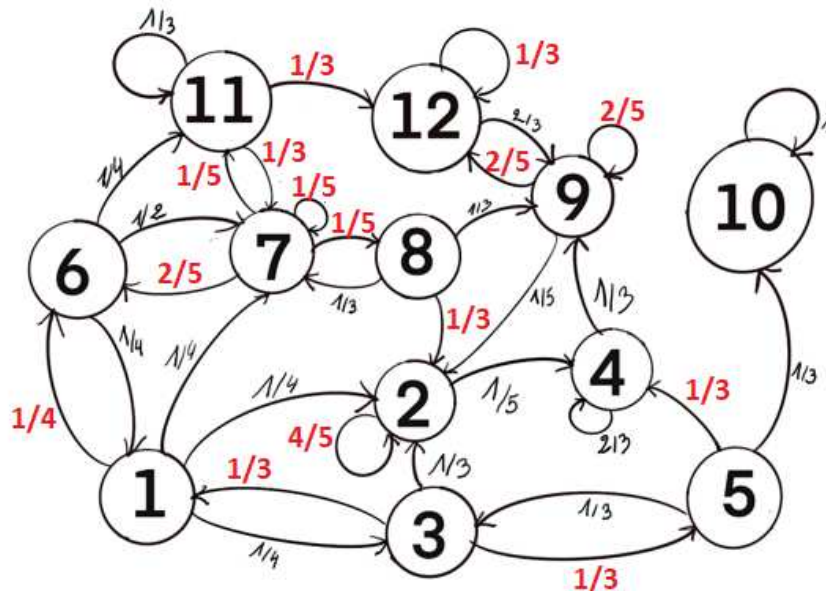
Nom :

Prénom :

- 2) Ce schéma correspond à un labyrinthe doté d'un système de portes. Celles marquées d'une flèche ne peuvent être passées que dans le sens de la flèche, les autres pouvant être passées dans les 2 sens. Un héros perdu dedans choisira aléatoirement une des portes s'offrant à lui. S'il essaie de passer une porte dans le mauvais sens, il va rester dans la pièce où il est avant de refaire son choix de la même manière.



- a. Complétez le diagramme de Markov ci-dessous.



- b. Partant du principe que la sortie est dans la salle n°10 et qu'au départ le héros est dans la salle n°1, est-il possible que le héros se retrouve bloqué sans pouvoir atteindre la sortie ? Si oui, dans quelle(s) salle(s) ? Justifiez en vous servant de la notion de classes de communication.

Il y a 3 classes de communication : $C_1 = \{1, 3, 5, 6, 7, 8, 11\}$, $C_2 = \{2, 4, 9, 12\}$ et $C_3 = \{10\}$. Or les classes C_2 et C_3 sont toutes les 2 récurrentes. Donc si le héros entre les salles 2, 4, 9 ou 12, il restera bloqué dans la classe C_2 sans jamais pouvoir en sortir et il n'atteindra jamais la sortie.

Nom :

Prénom :

- 3) Soit un jeu générant la matrice de Markov ci-dessous. On aimerait connaître le temps moyen pour atteindre l'état 5 en partant de l'état 1. Donnez le système, sans le résoudre, ainsi que l'inconnue à déterminer.

$$\begin{pmatrix} 1/6 & 1/5 & 1/8 & 0 & 1/10 \\ 1/6 & 0 & 1/8 & 1/2 & 2/5 \\ 1/3 & 3/5 & 1/4 & 1/6 & 0 \\ 1/3 & 1/5 & 0 & 1/6 & 0 \\ 0 & 0 & 1/2 & 1/6 & 1/2 \end{pmatrix}$$

Soit t_i le temps de parcours de l'état i à l'état 5. Alors le système à résoudre est le suivant :

$$\begin{cases} t_1 = 1 + \frac{1}{6}t_1 + \frac{1}{6}t_2 + \frac{1}{3}t_3 + \frac{1}{3}t_4 \\ t_2 = 1 + \frac{1}{5}t_1 + \frac{3}{5}t_3 + \frac{1}{5}t_4 \\ t_3 = 1 + \frac{1}{8}t_1 + \frac{1}{8}t_2 + \frac{1}{4}t_3 + \frac{1}{2}t_5 \\ t_4 = 1 + \frac{1}{2}t_2 + \frac{1}{6}t_3 + \frac{1}{6}t_4 + \frac{1}{6}t_5 \\ t_5 = 0 \end{cases}$$

Et la variable à déterminer est t_1 .

Nom :

Prénom :

Nom :

Prénom :

Question 3 (20 pts)

- a) Soit L le langage, sur l'alphabet $\Sigma = \{a, b, c\}$, formé de tous les mots qui comportent un nombre pair de a , sans jamais avoir deux a consécutifs et finissant par bc . Donnez une grammaire régulière engendrant ce langage ou un automate de Moore le reconnaissant. Précisez bien à quels mots correspond chaque symbole non terminal ou chaque état !

Grammaire :

$\langle S \rangle$: Nombre pair de a et ne finit ni par a , ni par b , ni par bc

$\langle A \rangle$: Nombre pair de a et finit par a

$\langle B \rangle$: Nombre pair de a et finit par b

$\langle C \rangle$: Nombre pair de a et finit par bc

$\langle D \rangle$: Nombre impair de a et ne finit pas par a

$\langle E \rangle$: Nombre impair de a et finit par a

$\langle S \rangle \rightarrow a\langle E \rangle \mid b\langle B \rangle \mid c\langle S \rangle$

$\langle A \rangle \rightarrow b\langle B \rangle \mid c\langle S \rangle$

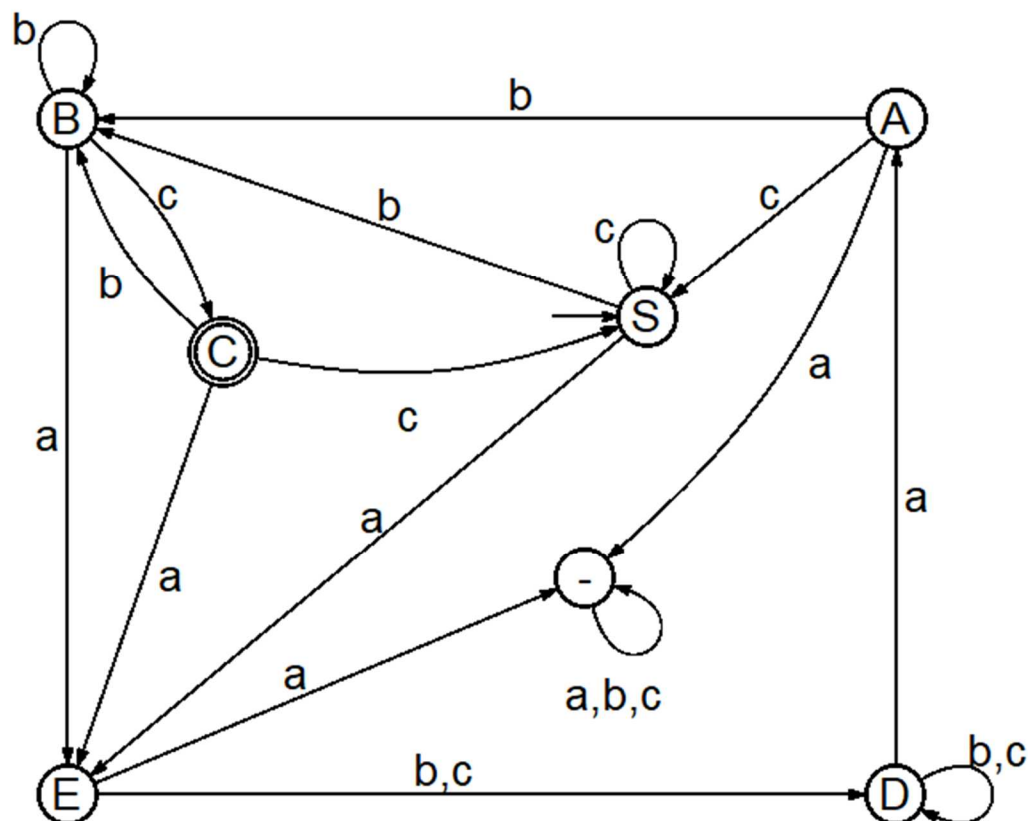
$\langle B \rangle \rightarrow a\langle E \rangle \mid b\langle B \rangle \mid c\langle C \rangle$

$\langle C \rangle \rightarrow \varepsilon \mid a\langle E \rangle \mid b\langle B \rangle \mid c\langle S \rangle$

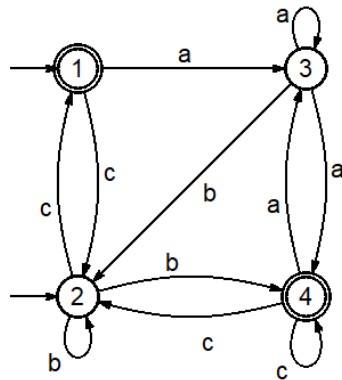
$\langle D \rangle \rightarrow a\langle A \rangle \mid b\langle D \rangle \mid c\langle D \rangle$

$\langle E \rangle \rightarrow b\langle D \rangle \mid c\langle D \rangle$

Automate de Moore :



Prénom :

$$M$$


The diagram illustrates a state transition system with the following states and transitions:

- States:** 1, 2, 3, 1,2,4, 2,4, 3,4, and -.
- Initial State:** 1,2 (indicated by an incoming arrow).
- Transitions:**
 - 1,2 to 3: labeled 'a'
 - 1,2 to 2,4: labeled 'b'
 - 1,2 to -: labeled 'c' (self-loop)
 - 3 to 1: labeled 'a'
 - 3 to 2: labeled 'c'
 - 3 to 1,2,4: labeled 'a'
 - 3 to 2,4: labeled 'c'
 - 1 to -: labeled 'b'
 - to 1: labeled 'c'
 - to 2: labeled 'a'
 - to 3: labeled 'c'
 - to 2,4: labeled 'b'
 - 2 to 1: labeled 'c'
 - 2 to 3: labeled 'a'
 - 2 to 1,2,4: labeled 'c'
 - 2 to 2,4: labeled 'b'
 - 1,2,4 to 1,2,4: labeled 'c' (self-loop)
 - 1,2,4 to 2,4: labeled 'b'
 - 2,4 to 1,2,4: labeled 'c'
 - 2,4 to 2,4: labeled 'b' (self-loop)
 - 2,4 to 3: labeled 'a'
 - 2,4 to 3,4: labeled 'c'
 - 3,4 to 3,4: labeled 'a' (self-loop)
 - 3,4 to 2,4: labeled 'b'
 - 3,4 to 2: labeled 'c'

Nom :

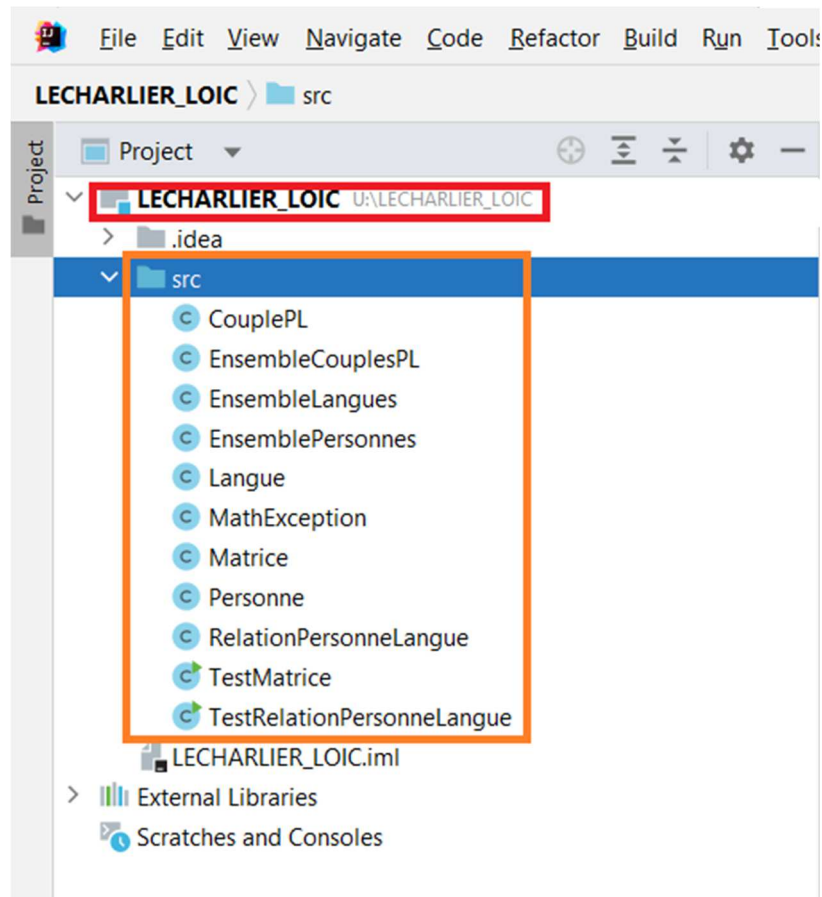
Prénom :

PARTIE II : IMPLÉMENTATION

Dans cette partie nous allons vous demander d'implémenter des méthodes dans plusieurs classes Java.

Pour ce faire :

- 1) Ouvrez IntelliJ
- 2) Créez, **sur le U** :, un projet NOM_PRENOM (**avec vos nom et prénom !**)
- 3) Les classes données se trouvent dans le répertoire « Classes Java ». Faites un copier-coller de celles-ci dans le répertoire « src » de votre projet IntelliJ. Voici ce que vous devriez obtenir :



Les questions 4 et 5 ci-après vous expliquerons ce que vous devez implémenter.

Nom :

Prénom :

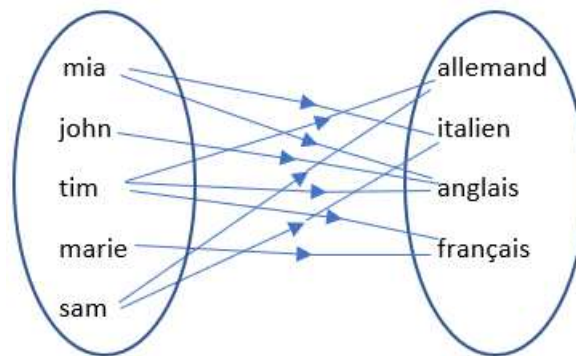
Question 4 (15 pts)

Un colloque international va réunir différentes personnes maîtrisant chacune une ou plusieurs langues.

Lors de son inscription, chaque personne a donné son nom, mais également une liste de langues qu'elle maîtrise. Chaque personne connaît au moins une langue.

Deux personnes peuvent communiquer entre elles si elles ont au moins une langue en commun.

Nous allons nous intéresser à la relation qui unit une personne à une langue :



Mia connaît l'italien et l'anglais, John connaît l'anglais, Tim connaît l'allemand, l'anglais et le français, ...

John et Marie sont unilingues.

Mia et Sam sont bilingues.

Tim est trilingue.

Mia peut communiquer avec John car ils connaissent tous les 2 l'anglais.

Elle peut aussi communiquer avec Tim et Sam, mais pas avec Marie.

...

Implémentation :

Voici les classes qui sont fournies (à ne pas modifier) : `Personne`, `Langue`, `CouplePL`, `EnsemblePersonnes`, `EnsembleLangues` et `EnsembleCouplesPL`.

Les différents ensembles possèdent les méthodes classiques sur les ensembles.

Ces ensembles peuvent être itérés :

```
for (Personne p : ensemblePersonnes)
```

```
for (Langue l : ensembleLangues)
```

```
for (CouplePL c : ensembleCouplesPL)
```

Vous allez compléter 2 méthodes de la classe `RelationPersonneLangue`.

Cette classe possède comme attributs 3 ensembles : `EnsemblePersonnes` + `EnsembleLangues` + `EnsembleCouplesPL`.

Elle possède un constructeur qui instancie ces 3 ensembles.

Elle contient les méthodes `estVide()`, `contient(CouplePL cpl)`,

`contient(Personne p, Langue l)`, `ajouter(Personne P, langue l)` et `ajouter(CouplePL cpl)`.

Les 2 méthodes à compléter sont : `bilingues()` et `peuventCommuniquer(Personne p1, Personne p2)`.

La méthode `bilingues()` renvoie l'ensemble des personnes qui sont bilingues. Ces personnes connaissent exactement 2 langues.

La méthode `peuventCommuniquer(Personne p1, Personne p2)` vérifie si les 2 personnes ont au moins une langue en commun.

La classe `TestRelationPersonneLangue` permet de tester les 2 méthodes à compléter.

```

74
75
76 //renvoie l'ensemble des personnes qui sont bilingues
77 //ces personnes connaissent exactement 2 langues (pas plus)!
78 public EnsemblePersonnes bilingues(){
79     //TODO
80     EnsemblePersonnes bil = new EnsemblePersonnes() ;
81     for (Personne p:this.ensemblePersonnes) {
82         int nbLangues = 0 ;
83         for (Langue la:this.ensembleLangues) {
84             if (this.contient(p,la)) {
85                 nbLangues++ ;
86                 if (nbLangues>2) {
87                     break ;
88                 }
89             }
90         }
91         if (nbLangues==2) {
92             bil.ajouter(p) ;
93         }
94     }
95     return bil ;
96 }
97
98 //verifie si les 2 personnes ont au moins une langue en commun
99 public boolean peuventCommuniquer(Personne p1, Personne p2){
100     if(p1==null || p2==null || !ensemblePersonnes.contient(p1) || !
ensemblePersonnes.contient(p2))
101         throw new IllegalArgumentException();
102     //TODO
103     for (Langue la:ensembleLangues) {
104         if (this.ensembleCouplesPL.contient(new CouplePL(p1,la))&&this.
ensembleCouplesPL.contient(new CouplePL(p2,la))) {
105             return true ;
106         }
107     }
108
109     return false;
110 }
111
112 public String toString(){
113     return ensemblePersonnes+ "\n"+ensembleLangues+ "\n"+ensembleCouplesPL;
114 }
115
116 }
117

```

Nom :

Prénom :

Question 5 (15 pts)

On vous demande d'implémenter la méthode

`int nbColonnesNulles()`

de la classe `Matrice`.

Cette méthode va renvoyer le nombre de colonnes nulles (de colonnes ne contenant que des 0) de la matrice courante.

Exemples :

1) Si $\text{this} = \begin{pmatrix} 0 & 0 & 3 \\ -1 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 1 car cette matrice n'a qu'une colonne nulle : la deuxième.

2) Si $\text{this} = \begin{pmatrix} 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & -4 & 0 & 4 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 3 car cette matrice a 3 colonnes nulles : la première, la deuxième et la quatrième.

3) Si $\text{this} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0.5 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -4 \\ -2 & 0 & 0 & 1 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 2 car cette matrice a 2 colonnes nulles : la deuxième et la troisième.

4) Si $\text{this} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0.5 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -4 \\ -2 & 0 & 0 & 1 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 0 car cette matrice n'a aucune colonne nulle.

5) Si $\text{this} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 5 car cette matrice a toutes ses colonnes qui sont nulles.

6) Si $\text{this} = \begin{pmatrix} 0 & -1 & 0 & 0.5 & 0 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 3 car cette matrice a 3 colonnes qui sont nulles : la première, la troisième et la cinquième.

7) Si $\text{this} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 1 car cette matrice a son unique colonnes qui est nulle.

8) Si $\text{this} = \begin{pmatrix} 0 \\ 0 \\ -1.5 \\ 0 \end{pmatrix}$ alors la méthode `nbColonnesNulles()` va renvoyer 0 car cette matrice n'a pas de colonne nulle.

```

1  import java.util.Arrays ;
2  public class Matrice {
3      private final int nbLignes;           // nombre de lignes
4      private final int nbColonnes;         // nombre de colonnes
5      private final double[][] data;        // matrice (nbLignes,nbColonnes)
6
7      // ce constructeur cree la matrice nulle de genre (a,b)
8      public Matrice(int a, int b) throws IllegalArgumentException {
9          if (a<=0 || b<=0)
10             throw new IllegalArgumentException("a ou b négatif") ;
11          data = new double[a][b] ;
12          nbLignes = a ;
13          nbColonnes = b ;
14      }
15
16      // Renvoie le nombre de colonne nulle (ne contenant que des 0)
17      // Exemple : voir énoncé
18      public int nbColonnesNulles() {
19          // TODO
20          int nbColNulles = 0 ;
21          for (int j=0 ; j<nbColonnes ; j++) {
22              int i = 0 ;
23              while (i<nbLignes && data[i][j]==0) {
24                  i++ ;
25              }
26              if (i==nbLignes) {
27                  nbColNulles++ ;
28              }
29          }
30          return nbColNulles ;
31      }
32
33      // affiche la matrice en format standard //NE PAS MODIFIER CETTE METHODE !!!
34      public String toString(){
35          String st = "" ;
36          int tmax = 0 ;
37          for (int i=0 ; i<nbLignes ; i++) {
38              for (int j=0 ; j<nbColonnes ; j++) {
39                  String s = "" + data[i][j] ;
40                  if (data[i][j]>=0)
41                      s = " "+s ;
42                  if (s.length()>tmax)
43                      tmax = s.length() ;
44              }
45          }
46          for (int i=0 ; i<nbLignes ; i++) {
47              for (int j=0 ; j<nbColonnes ; j++) {
48                  String s = "" + data[i][j] ;
49                  if (data[i][j]>=0)
50                      s = " "+s ;
51                  st = st + s ;
52                  int nbBlanc = tmax-s.length()+2;
53                  for (int k=0 ; k<nbBlanc ; k++)
54                      st = st + " " ;
55              }
56              if (i<nbLignes-1)
57                  st = st+'\n' ;
58          }
59          return st ;
60      }
61  }
62  }
63

```