

**Examen de Mathématiques 2 :**

1ère 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** |

**PARTIE I : SUR PAPIER**

**Question 1 (20 pts)**

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

h

g

f

e

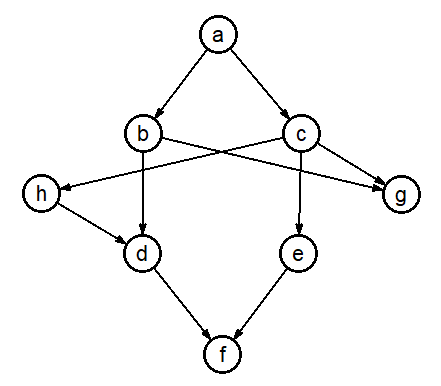
d

c

a

b

1. Complétez le digraphe qui suit pour qu’il représente la même relation :



1. Pour chacun des ordres totaux ci-dessous, indiquez « oui » si c’est un tri topologique de ≤, et « non » sinon. Si c’est « non », justifiez votre assertion.
   1. : a → b → c → g → e → h → d → f
   2. : a → b → h → c → e → d → g → f
   3. : a → c → g → e → b → h → d → f

1. L’ordre partiel ≤ sur E est-il un treillis ? Justifiez votre réponse.
2. Indiquez si ≤ sur E possède un minimum, un maximum, un infimum et un suprémum. Dans l’affirmative, donnez l’élément. Dans la négative, justifiez.
3. Relativement à ≤ sur E, déterminez les 4 ensembles suivants : les minimaux, les maximaux, les minorants et les majorants.
4. Proposez un sous-ensemble de E possédant au moins 5 éléments et qui soit un treillis pour ≤.

**Question 2 (20 pts)**

1. Le Salsyn est un jeu traditionnel qui se joue avec un dé à dix faces. Chaque joueur démarre la manche avec 5 jetons. A son tour, il lance le dé. S’il fait 2, il perd un jeton. S’il fait un 1, il perd deux jetons (sans pouvoir descendre sous 0 jeton). En revanche, s’il fait un 10, il regagne 2 jetons (sans pouvoir grimper au-dessus de 5 jetons). Le jeu se termine quand le joueur n’a plus de jeton.
   1. Complétez le diagramme du processus de Markov et écrivez la matrice de transition.

0

1

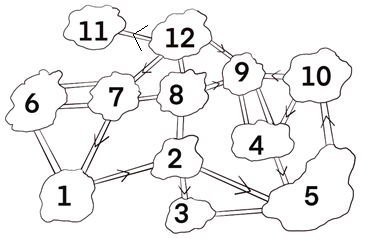
5

2

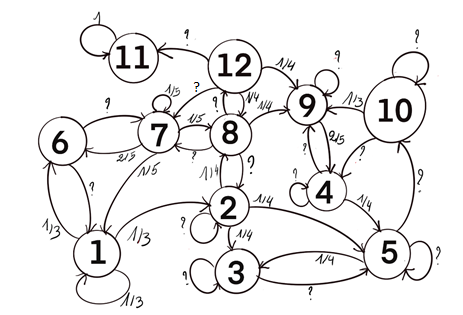
4

3

1. 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. Elles sont symbolisées par une flèche > . 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.



* 1. Complétez le diagramme de Markov ci-dessous.



* 1. Partant du principe que la sortie est dans la salle n°11 et que le héros est dans la salle n°1, est-il possible que le héros se retrouve bloquer sans pouvoir atteindre la sortie ? Si oui, dans quelle(s) salle(s) ? Justifiez en vous servant de la notion de classe de communication.

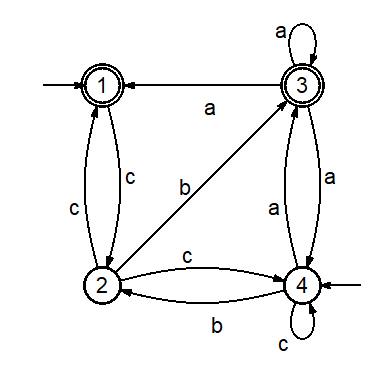
1. Soit un jeu générant la matrice de Markov ci-dessous. On aimerait connaître le temps moyen pour atteindre l’état 5 en partant de l’état 1. Donnez le système, sans le résoudre, ainsi que l’inconnue à déterminer.

**Question 3 (20 pts)**

1. Soit L le langage, sur l’alphabet , formé de tous les mots qui comportent un nombre impair de b et dans lesquels chaque a est toujours suivi ou précédé par un autre a. Donnez une grammaire régulière engendrant ce langage ou un automate de Moore le reconnaissant. Précisez bien à quel mot correspond chaque symbole non terminal ou chaque état !

|  |
| --- |
|  |

1. Soit le langage défini sur reconnu par le NDFA .

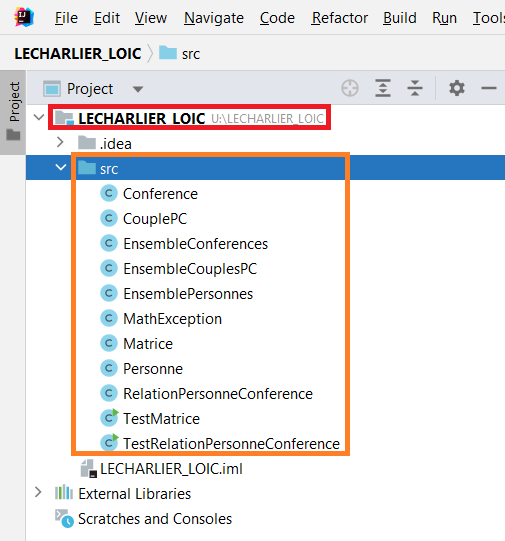


Utilisez la subset construction afin d’obtenir un automate de Moore reconnaissant le langage  :

**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.

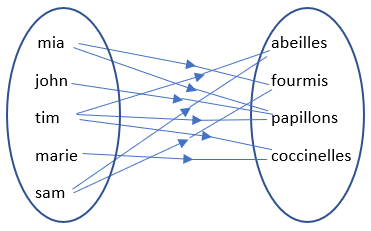
**Question 4 (15 pts)**

Plusieurs conférences sont proposées pendant toute la durée d’une exposition sur les insectes.

Lors de son inscription, chaque personne a donné son nom, mais également une liste des conférences auxquelles elle va assister. Chaque personne s’inscrit à au moins une conférence.

Deux personnes vont se croiser si elles se sont inscrites à au moins une même conférence.

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



Mia s’est inscrite à la conférence sur les fourmis et à celle sur les papillons.

John s’est inscrit à la conférence sur les papillons.

Tim s’est inscrit à la conférence sur les abeilles, à celle sur les papillons et à celle sur les coccinelles.

…

John et Marie se sont inscrits à 1 conférence.

Mia et Sam se sont inscrits à 2 conférences.

Tim s’est inscrit à 3 conférences.

Mia va croiser John car ils sont tous les 2 inscrits à la conférence sur les papillons.

Elle croisera également Tim et Sam, mais pas Marie.

…

**Implémentation :**

Voici les classes qui sont fournies (à ne pas modifier) : Personne, Conference, CouplePC, EnsemblePersonnes, EnsembleConference et EnsembleCouplesPC.

Les différents ensembles possèdent les méthodes classiques sur les ensembles.

Ces ensembles peuvent être itérés :

for(Personne p : ensemblePersonnes)

for(Conference c : ensembleConferences)

for(CouplePC cpc : ensembleCouplesPC)

Vous allez compléter 2 méthodes de la classe RelationPersonneConference.

Cette classe possède comme attributs 3 ensembles : EnsemblePersonnes + EnsembleConferences + EnsembleCouplesPC.

Elle possède un constructeur qui instancie ces 3 ensembles.

Elle contient les méthodes estVide(), contient(CouplePC cpc), contient(Personne p, Conference c), ajouter(Personne p, Conference c)et ajouter(CouplePC cpc).

**Les 2 méthodes à compléter sont : inscritesA2Conferences()et vontSeCroiser(Personne p1, Personne p2).**

La méthode inscritesA2Conferences() renvoie l’ensemble des personnes qui sont inscrites à exactement 2 conférences.

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

La classe TestRelationPersonneConference permet de tester les 2 méthodes à compléter.

**Question 5 (15 pts)**

On vous demande d’implémenter la méthode

int nbLignesNonNulles()

de la classe Matrice.

Cette méthode va renvoyer le nombre de lignes non nulles de la matrice courante c-à-d le nombre de ligne ayant au moins un élément différent de 0.

*Exemples :*

1. Si this = alors la méthode nbLignesNonNulles() va renvoyer 1 car cette matrice n’a qu’une ligne non nulle : la deuxième.
2. Si this = alors la méthode nbLignesNonNulles() va renvoyer 3 car cette matrice a 3 lignes non nulles : la première, la troisième et la quatrième.
3. Si this = alors la méthode nbLignesNonNulles() va renvoyer 2 car cette matrice a 2 lignes non nulles : la première et la troisième.
4. Si this = alors la méthode nbLignesNonNulles() va renvoyer 5 car cette matrice a toutes ses lignes qui sont non nulles.
5. Si this = alors la méthode nbLignesNonNulles() va renvoyer 0 car cette matrice n’a aucune ligne non nulle.
6. Si this = alors la méthode nbLignesNonNulles() va renvoyer 0 car cette matrice n’a aucune ligne non nulle.
7. Si this = alors la méthode nbLignesNonNulles() va renvoyer 1 car cette matrice a son unique ligne non nulle.
8. Si this = alors la méthode nbLignesNonNulles() va renvoyer 2 car cette matrice a 2 lignes non nulles : la deuxième et la troisième.

**BROUILLON**

**BROUILLON**

**BROUILLON**