

NOM :

Août 2022

PRÉNOM :

BLOC :



Examen de Mathématiques 1 :

1^{ère} année Bachelier en Informatique de Gestion

BINV1090 – Mathématiques 1

Date : 25 août 2022

Durée de l'examen : 3 heures

Nombre de questions : 6

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	/10
Question 2	/10
Question 3	/10
Question 4	/15
Question 5	/15
Question 6	/20
TOTAL	/80

Nom :

Prénom :

Question 1 (10 pts)

- 1) Sur l'île de Puro-Pira, il n'y a que 2 types d'habitants : les Purs qui ne disent que des propositions vraies et les Pires qui ne disent que des propositions fausses.

Vous croisez 3 habitants A, B et C :

- A dit « C et moi sommes des Pires »
- B dit « Si C est un Pur alors A aussi »
- C dit « Nous ne sommes pas tous des Purs »

De quel type sont A, B et C ? **Justifiez !**

- 2) Soient les deux formules ci-dessous.

- $\neg b \Rightarrow [(\neg a \vee c) \Rightarrow (a \wedge c)]$
- $\neg a \wedge \neg b$

Ces deux formules sont-elles équivalentes, la négation l'un de l'autre ou ni l'un ni l'autre ?
Justifiez !

Nom :

Prénom :

Question 2 (10 pts)

Dans l'univers des personnes, on considère les prédicats suivants :

- $hom(x)$ signifie x est un homme
- $fem(x)$ signifie x est une femme
- $marie(x,y)$ signifie x est marié à y
- $aime(x,y)$ signifie x aime y

On considère aussi la constante C désignant une personne appelée Camille.

- a) Codez la phrase « Camille est le seul homme qui peut être marié à plus d'une femme » ;

- b) Décodez, de la manière la plus élégante possible, la formule suivante :

$$\forall x \left[[hom(x) \wedge \exists y \exists z (fem(y) \wedge fem(z) \wedge aime(x,y) \wedge aime(x,z) \wedge \neg(y = z))] \Rightarrow \exists u marie(x,u) \right]$$

Nom :

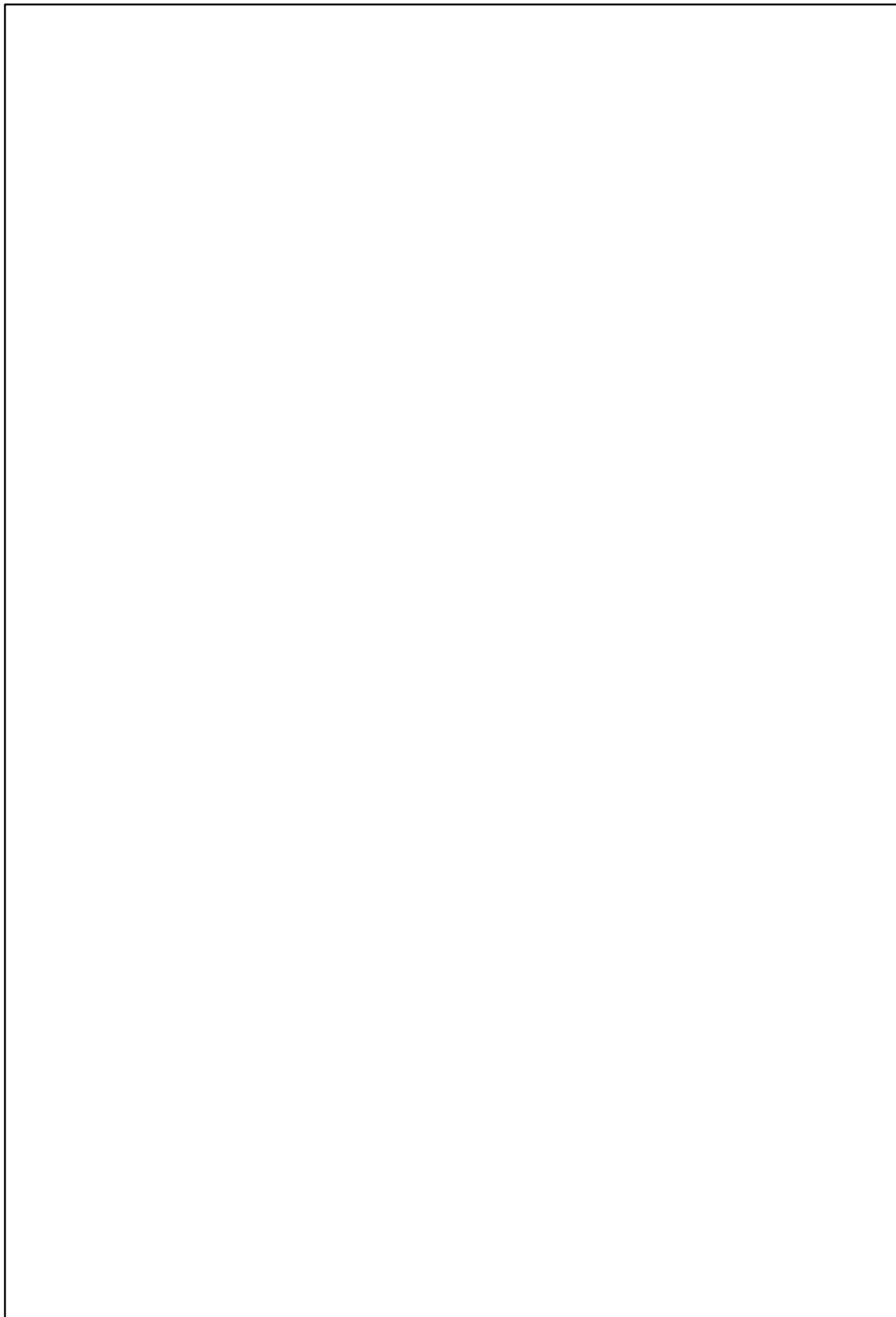
Prénom :

Question 3 (10 pts)

Démontrez par récurrence que $2^{4n} - 1$ est divisible par 15 pour tout naturel n .

Nom :

Prénom :

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed response.

Nom :

Prénom :

Nom :

Prénom :

Question 4 (15 pts)

a) Soient les ensembles $A = \{1,2,3\}$ et $B = \{|A|, A\}$

i) Donnez B en extension.

ii) Donnez $P(A \oplus B) - (P(A) \cup P(B))$ en extension. Justifiez !

b) Donnez un ensemble X tel que $P(X) - X \neq P(X)$. Justifiez !

c) Soient A, B, C des sous-ensembles d'un Univers E de cardinal 6. Sur le diagramme de Venn ci-contre, les minuscules a, b, c, \dots, h désignent les cardinaux des zones correspondantes.

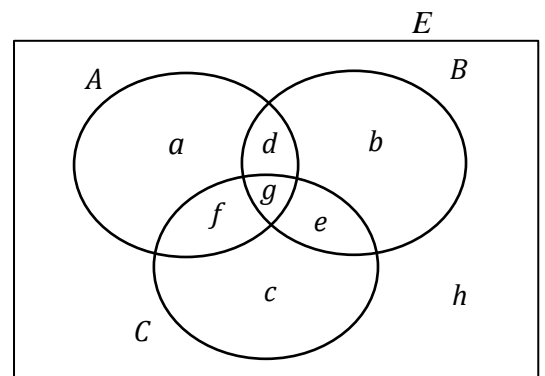
i) Commencez par traduire chacune des 4 affirmations suivantes par une information concernant les entiers a, b, c, \dots

1° $|A| < |B| < |C|$

2° $A \oplus (B \cap \bar{C}) = B - A$

3° $|C - (A \oplus B)| = |P(\{A \cup B\}, 2)|$

4° $A \oplus \bar{B} \neq (A \cap B) \cup (C - A)$



Réponse. Traduction des informations 1° à 4° (vous ne devez pas justifier) :

1° :

2° :

3° :

4° :

Nom :

Prénom :

ii) En supposant vraies les 4 informations du point i), que pouvez-vous dire à propos de $|B|$?
Justifiez votre réponse !

Nom :

Prénom :

Question 5 (15 pts)

Voici une proposition de méthode itérative pour calculer la racine cubique du réel $a \geq 1$:

$$\begin{cases} x_0 = 1 \\ x_{n+1} = \frac{(a-1)x_n + \frac{a}{x_n^2}}{a} \end{cases}$$

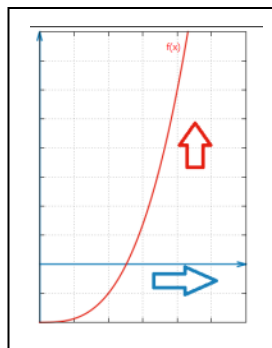
En Excel, calculez les 10 premières approximations de la méthode proposée et celles de la méthode de Newton (vue au cours) appliquée sur l'intervalle $[1, a]$.

Que constatez-vous ? Tirez vos conclusions.

Complétez le fichier Excel *Newton_Prop*.

Remarques :

- La valeur choisie du réel dont on veut connaître la racine est 10, mais cette valeur peut être modifiée par n'importe quel réel strictement positif.
- Pour Newton, déduisez l'approximation initiale en observant le graphique suivant qui présente l'allure de la fonction $f(x) = x^3 - a$.



- $f'(x) = 3x^2 - 2$
- Pour toute éventuelle colonne ajoutée, n'oubliez pas de lui donner un titre.
- Toute valeur éventuelle provenant d'un calcul isolé doit aussi être décrite.

Rappel :

- La formule de récurrence de Newton est :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

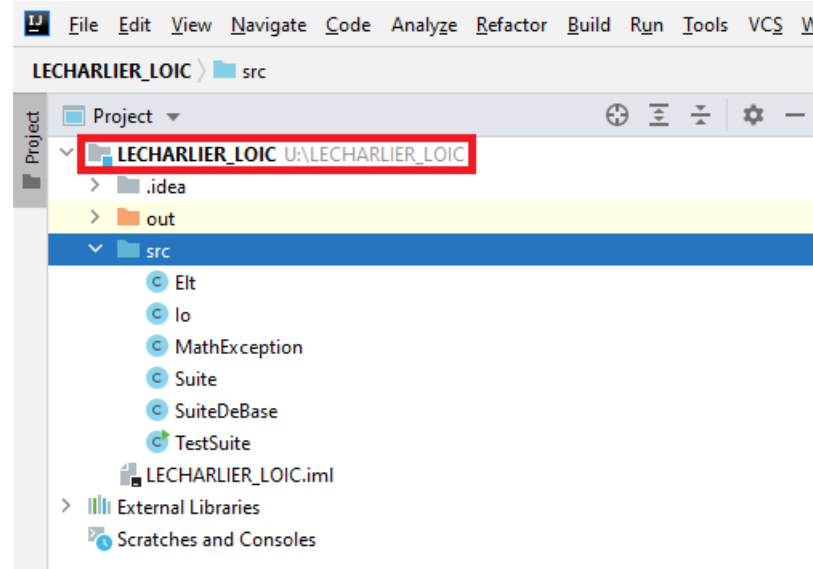
Nom :

Prénom :

Question 6 (20 pts)

On vous demande de compléter 2 méthodes de la classe `Suite`, « héritant » de la classe `SuiteDeBase`. 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 :



On vous demande de programmer **au moins une des deux méthodes** ci-dessous en utilisant la technique **récursive**.

Vous pouvez utiliser toutes les méthodes qui apparaissent dans le document joint "Memento_Suite_Java.pdf".

Si vous utilisez d'autres méthodes, vous devez donner leur code.

Vous pouvez tester vos solutions grâce à la classe `TestSuite`.

Attention ! Il est interdit d'introduire d'autres méthodes, exceptée les méthodes privées ayant les mêmes paramètres que les méthodes publiques, dans le cas d'une version récursive où il y a des exceptions à gérer.

Méthode 1 : `aucunEntre(int a, int b)`

```
/* Renvoie true si aucun élément de la suite n'est compris entre a et b
 *      false sinon
 * Précondition à ne pas vérifier : a<=b
 * Exemples :
 * -----
 * this = (1,5,6,10,3) alors aucunEntre(6,9)    --> false
 * this = (1,5,6,10,3) alors aucunEntre(7,9)    --> true
 * this = (1,5,6,10,3) alors aucunEntre(-5,-2)  --> true
 * this = (1,5,6,10,3) alors aucunEntre(-2,5)   --> false
 * this = ()      alors aucunEntre(6,9)         --> true
 * @param int a : borne inférieure
 * @param int b : borne supérieure
 * @return true si aucun élément de la suite courante n'est compris
 *          entre a et b
 *          false sinon
 */
```

Nom :

Prénom :

Méthode 2 : auMoinsNOccurences (int n, Elt e)

```
/* Renvoie true si la suite contient au moins n fois l'Elt e et
 *      false sinon
 * Exemples :
 * -----
 * this = (3,7,7,12,7), n=2 et e=7      alors
 *      auMoinsNOccurences(n,e) --> true
 * this = (3,7,7,12,7), n=3 et e=7      alors
 *      auMoinsNOccurences(n,e) --> true
 * this = (3,7,7,12,7), n=4 et e=7      alors
 *      auMoinsNOccurences(n,e) --> false
 * this = (), n=0 et e=2      alors
 *      auMoinsNOccurences(n,e) --> true
 * this = (), n=1 et e=7      alors
 *      auMoinsNOccurences(n,e) --> false
 * this = (3,7,7,12,7), n=-1 et e=7      alors
 *      auMoinsNOccurences(n,e) --> IllegalArgumentException
 * this = (3,7,7,12,7), n=-8 et e=7      alors
 *      auMoinsNOccurences(n,e) --> IllegalArgumentException
 * this = (3,7,7,12,7), n=1 et e=null    alors
 *      auMoinsNOccurences(n,e) --> IllegalArgumentException
 *
 * @param int n : nombre minimum d'occurences de l'Elt e
 * @param Elt e : élément de la suite dont on cherche n occurrence(s)
 * @return true si la suite contient au moins n fois l'Elt e
 *      false sinon
 * @throw IllegalArgumentException si n<0
 * @throw IllegalArgumentException si e est null
 */
```

Nom :

Prénom :
BROUILLON

Nom :

Prénom :
BROUILLON

Nom :

Prénom :
BROUILLON

Nom :

Prénom :
BROUILLON