NOM :

PRÉNOM :

BLOC :

Janvier 2022

Haute École Léonard de Vinci
Sciences et Techniques
Informatique de gestion *Paul Lambin*

Examen de Mathématiques 1 :

1ère année Bachelier en Informatique de Gestion

BINV1090 – Mathématiques 1

Date : 18 janvier 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**

Question 1 (10 pts)

Soit la formule $\big((a \wedge b) \vee c\big) \Rightarrow (a \vee \neg b)$.

a) Simplifiez cette formule le plus possible. Donnez-en une formulation équivalente où n'apparaissent ni parenthèses, ni symbole d'implication, ni le symbole d'équivalence et où le signe de négation ne porte que sur des propositions

b) Parmi les formules ci-dessous, quelle(s) est (sont) celle(s) à la formule ci-dessus. Justifiez.

- $\bar{c} + a + \bar{b}$
- $c \Rightarrow (a \vee \neg b)$
- $b \Rightarrow (a \vee c)$

Question 2 (10 pts)

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

- *fem(x)* signifie x est une femme
- *hom(x)* signifie x est un homme
- *membre(x,y)* signifie x est membre de y
- *club(x)* signifie x est un club privé

On considère aussi la constante C désignant le Country club de Bruxelles
Remarque : on considère qu'une personne est soit un homme soit une femme

a) Codez la phrase « Un membre du country club de Bruxelles ne peut pas être membre d'un autre club privé sauf si c'est une femme » ;

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

$club(C) \wedge (\forall x\ [club(x) \wedge\ \exists y \exists z\ (membre(y,x) \wedge fem(y) \wedge membre(z,x) \wedge fem(z) \wedge y \neq z)] \Rightarrow (x \neq C))$

Question 3 (10 pts)

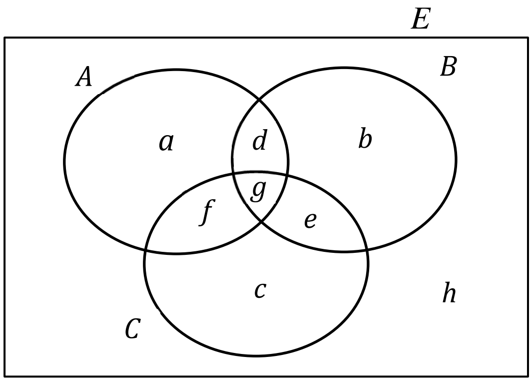
Démontrez par récurrence que

$$1 + 4 + 7 + \cdots + (3n - 2) = \frac{3n^2 - n}{2}$$

Pour tout naturel $n \geq 1$.

Question 4 (15 pts)

Soient A, B, C des sous-ensembles d'un Univers E de cardinal 7. Sur le diagramme de Venn ci-contre, les minuscules a , b , c , … , h désignent les cardinaux des zones correspondantes.



a) Commencez par traduire chacune des 5 affirmations suivantes par une information concernant les entiers a , b , c , …

1° $\left|\overline{\bar{A} \oplus B \oplus \bar{C}}\right| = \left|P\left(\{1, \{3,7\}\} \oplus \{3,1\}\right)\right|$

2° $|A \cup B| = |A| + |B|$

3° $(B \cup A) - (A - C) = \overline{E - \left((A \oplus B) - (C - A)\right)}$

4° $|C| < |B|$

5° $(B \cup \bar{A}) \cap C \nsubseteq A \oplus B$

Réponse. Traduction des informations 1° à 5° : (Vous ne devez pas justifier)

1° :

2° :

3° :

4° :

5° :

b) En supposant vraies les 5 informations du point a), donnez $\left|\overline{A \cup B \cup C}\right|$. Justifiez votre réponse !

7/14

Question 5 (15 pts)

La méthode de Newton modifiée est une méthode numérique de recherche d'une racine d'une fonction.
C'est une variante de la méthode de Newton vue au cours.
Elles possèdent les mêmes conditions de convergence.
Le choix de l'approximation initiale x_0 est pareil.
Le calcul de la première approximation est identique : on prend la tangente à la fonction en x_0. En prenant l'intersection entre cette droite et l'axe des abscisses, on obtient une nouvelle approximation x_1.
Pour Newton, on réitère ce principe de la tangente pour toutes les approximations suivantes.
Pour Newton modifié, on prend chaque fois une droite parallèle à la 1ère (et unique) tangente tracée.

La formule de récurrence de Newton est :

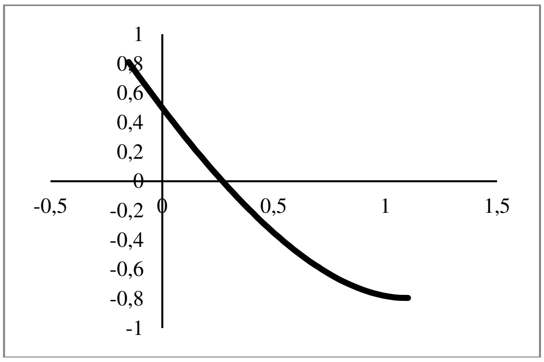
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

La formule de récurrence de Newton modifié est :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

La méthode de Newton modifiée demande plus d'étapes que la méthode de Newton pour obtenir la racine avec une même précision.
Mais le calcul de l'approximation est beaucoup plus rapide. **Il ne faut calculer qu'une seule fois « la tangente ».**

Considérons la fonction $f(x) = e^x - 3x - 0.5$ dont le graphe est donné ci-dessous.



En Excel, calculez les 7 premières approximations de la méthode de Newton et de la méthode de Newton modifiée appliquées sur l'intervalle [0,1].
Complétez la feuille du ficher Excel Newton_NewtonModifie qui se trouve sur le U : et **Laissez-le bien à cet endroit !**

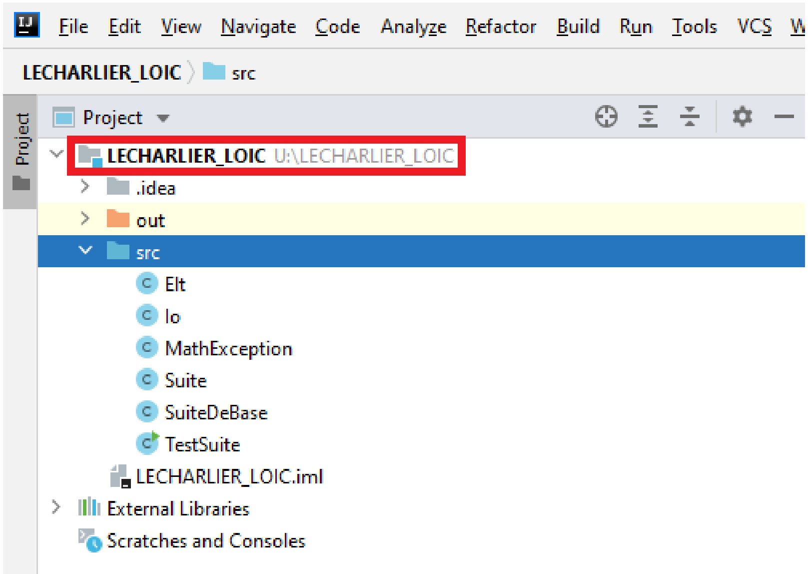
Remarques :

- Vous pouvez entrer directement le point variable. Déduisez-le en observant le graphique.
- $f'(x) = e^x - 3$
- En Excel, la fonction e^x est donnée par la fonction EXP.
- 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.

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 les 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`.

Méthode 1 : `tousDiviseursDe(int x)`

```
/** Renvoie true si tous les éléments de la suite courante sont des
 *                 diviseurs de x
 *            false sinon
 * Exemples :
 * ----------
 * this = (3,9,6,3,10) alors tousDiviseursDe(36)   --> false
 * this = (3,9,6,12,3) alors tousDiviseursDe(36)   --> true
 * this = ()           alors tousDiviseursDe(36)   --> true
 * this = (8,2,6)      alors tousDiviseursDe(-24)  --> true
 * this = (8,3,6)      alors tousDiviseursDe(0)    --> true
 * @param int x
 * @return true si tous les éléments de la suite courante sont des
 *                 diviseurs de x
 *            false sinon
 **/
```

Méthode 2 : `ajouterEnKEmePosition(Elt e, int k)`

```
/** Renvoie une copie de suite courante dans laquelle on a ajouté
 *  l'Elt e à la position k
 *  Exemples :
 *  ----------
 *  this = (1,7,6) ;  e = new Elt(3) et k = 2  alors
 *         ajouterEnKEmePosition(e, k) --> (1,3,7,6)
 *  this = (1,7,6) ;  e = new Elt(3) et k = 4  alors
 *         ajouterEnKEmePosition(e, k) --> (1,7,6,3)
 *  this = (1,7,6) ;  e = new Elt(3) et k = 5  alors
 *       ajouterEnKEmePosition(e, k) --> IllegalArgumentException
 *  this = (1,7,6) ;  e = new Elt(3) et k = -1 alors
 *       ajouterEnKEmePosition(e, k) --> IllegalArgumentException
 *  this = ()      ;  e = new Elt(3) et k = 1  alors
 *         ajouterEnKEmePosition(e, k) --> (3)
 * @param Elt e
 * @param int k
 * @return une suite qui est une copie de la suite courante dans
 *         laquelle on a ajouté l'Elt e à la position k
 * @throw IllegalArgumentException si e est null
 * @throw IllegalArgumentException s'il n'est pas possible
 *        d'ajouter l'Elt e à la position k
 */
```

BROUILLON

BROUILLON

BROUILLON

BROUILLON