

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Réussir en informatique

#### Connaître ses faiblesses pour mieux y répondre

DE KINDER, Jeremy; PEYTIER, Paul

*Award date:*  
2023

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



**UNIVERSITÉ  
DE NAMUR**

FACULTÉ  
D'INFORMATIQUE

UNIVERSITÉ DE NAMUR  
Faculté d'informatique  
Année académique 2022-2023

**Réussir en informatique : Connaître  
ses faiblesses pour mieux y répondre**

DE KINDER Jeremy  
PEYTIER Paul

..... (Signature pour approbation du dépôt - REE art. 40)

Promoteur : SCHUMACHER Laurent

Co-promoteur : HENRY Julie

Mémoire présenté en vue de l'obtention du grade de Master 60 en Sciences Informatiques

Faculté d'Informatique – Université de Namur

RUE GRANDGAGNAGE, 21 ● B-5000 NAMUR(BELGIUM)

# Remerciements

Nous tenons à remercier notre promotrice, Madame Julie HENRY, Cheffe de projet STEAM à l'Université de Namur, ainsi que notre promoteur, Monsieur Laurent SCHUMACHER, Professeur à l'Université de Namur, pour leurs précieux conseils et soutiens apportés durant l'élaboration de notre mémoire.

Nous tenons aussi à remercier Monsieur Jérôme FINK, assistant à la Faculté d'informatique de l'Université de Namur, et Madame Isabelle DAELMAN, secrétaire administrative à la Faculté d'informatique de l'Université de Namur, pour l'aide apportée pour la transmission des questionnaires ainsi que Monsieur Gerry LONGFILS, étudiant à l'Université de Namur, pour la prise de contact concernant l'outil en ligne Inginious.

Nous remercions également nos familles et amis respectifs pour le soutien moral, la motivation et les encouragements qu'ils nous ont donnés durant ces années d'études.

# Résumé

L'utilisation du facteur d'auto-évaluation permet aux étudiants de prendre conscience de leurs connaissances et compétences en se positionnant par rapport à des questions spécifiques.

Le but de cette recherche est, dans un premier temps, d'utiliser ce facteur dans le cadre de questionnaires techniques en mathématiques et en programmation à destination des étudiants inscrits au cours d'introduction à la programmation afin d'identifier leurs faiblesses dans ces matières.

Nos résultats montrent que l'auto-évaluation permet de mettre en avant des lacunes au niveau de ces matières et de déterminer les capacités d'auto-estimation des étudiants.

Dans un second temps, nous explorons différentes améliorations possibles pour l'utilisation du facteur d'auto-évaluation et qui pourraient faire l'objet de recherches futures.

*Mots-clés : auto-évaluation, degrés de certitude, facteur de réussite, introduction à la programmation, psychologie de la programmation*

# Abstract

The use of the self-assessment factor enables students to become aware of their knowledge and skills by positioning themselves in relation to specific questions.

The aim of this research is, firstly, to use this factor in the context of technical questionnaires in mathematics and programming aimed at students enrolled on the introductory programming course in order to identify their weaknesses in these subjects.

Our results show that self-assessment makes it possible to highlight shortcomings in these subjects and to determine students' self-assessment abilities.

Secondly, we explore possible improvements to the use of the self-assessment factor that could be the subject of future research.

*Keywords : self-assessment, degrees of certainty, success factor, introduction of programming, psychology of programming*

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>État de l'art</b>	<b>9</b>
2.1	Contexte général . . . . .	9
2.1.1	Identification précoce des étudiants à risque . . . . .	10
2.1.2	Utilisation d'outils numériques . . . . .	11
2.1.3	Performance des étudiants . . . . .	11
2.1.4	Passeport pour le bac . . . . .	12
2.2	L'auto-efficacité . . . . .	13
2.3	L'auto-évaluation . . . . .	14
2.4	Auto-efficacité et auto-évaluation, deux facteurs identiques? . . . . .	15
2.5	Matières visées par la recherche . . . . .	15
2.6	Centration et degrés de certitude . . . . .	16
2.7	La psychologie de la programmation . . . . .	18
<b>3</b>	<b>Problématique</b>	<b>20</b>
3.1	Positionnement de la recherche . . . . .	20
3.2	Questions de recherche . . . . .	20
3.2.1	QR1 : Quelles faiblesses dans l'apprentissage de la programmation l'auto-évaluation permet-elle d'identifier? . . . . .	21
3.2.2	QR2 : Quelles améliorations peuvent être envisagées suite à l'intégration de l'auto-évaluation dans l'apprentissage de la programmation? . . . . .	21
<b>4</b>	<b>Méthodologie</b>	<b>22</b>
4.1	Population visée . . . . .	22
4.2	Création des questionnaires . . . . .	22
4.2.1	Création des premiers questionnaires . . . . .	23
4.2.2	Création du questionnaire post-examen . . . . .	26
4.3	Passation des questionnaires . . . . .	26
4.3.1	Les premiers questionnaires . . . . .	26
4.3.2	Le questionnaire post-examen . . . . .	28
4.4	Collecte des résultats . . . . .	29
4.4.1	Les premiers questionnaires . . . . .	29
4.4.2	Le questionnaire post-examen . . . . .	30
<b>5</b>	<b>Résultats</b>	<b>31</b>
5.1	Questionnaires d'auto-évaluation . . . . .	31
5.1.1	Connaissances en algèbre . . . . .	31
5.1.2	Connaissances en mathématiques discrètes . . . . .	32

5.1.3	Niveau d'anglais . . . . .	33
5.1.4	Connaissances en programmation Python . . . . .	34
5.1.5	Connaissances d'autres langages de programmation . . . . .	34
5.2	Questionnaire de mathématiques . . . . .	35
5.2.1	Identification de la matière posant problème . . . . .	36
5.2.2	La centration . . . . .	37
5.3	Questionnaire de programmation en langage Python . . . . .	39
5.3.1	Identification de la matière posant problème . . . . .	40
5.3.2	La centration . . . . .	41
5.4	Questionnaire post-examen . . . . .	43
5.4.1	Etudiants ayant complété les questionnaires précédents . . . . .	44
5.4.2	Etudiants n'ayant pas complété les questionnaires précédents . . . . .	44
<b>6</b>	<b>Discussion</b>	<b>46</b>
6.1	QR1 : Quelles faiblesses dans l'apprentissage de la programmation l'auto-évaluation permet-elle d'identifier? . . . . .	46
6.1.1	Matières problématiques . . . . .	46
6.1.2	Centration et auto-évaluation . . . . .	49
6.1.3	Faiblesses identifiées . . . . .	49
6.2	QR2 : Quelles améliorations peuvent être envisagées suite à l'intégration de l'auto-évaluation dans l'apprentissage de la programmation? . . . . .	49
6.2.1	Création d'un outil et/ou de modules d'apprentissage - Inginious . . . . .	50
6.2.2	Aide à destination des professeurs ou des assistants . . . . .	50
6.2.3	Feedback vers les étudiants . . . . .	50
<b>7</b>	<b>Biais</b>	<b>52</b>
<b>8</b>	<b>Conclusion</b>	<b>54</b>
<b>9</b>	<b>Bibliographie</b>	<b>55</b>
	<b>Annexes</b>	<b>59</b>
<b>A</b>	<b>Questionnaire d'auto-évaluation</b>	<b>59</b>
<b>B</b>	<b>Questionnaire de mathématiques</b>	<b>63</b>
<b>C</b>	<b>Questionnaire de programmation en langage Python</b>	<b>88</b>
<b>D</b>	<b>Questionnaire post-examen</b>	<b>103</b>
D.1	L'étudiant a complété les questionnaires . . . . .	104
D.2	L'étudiant n'a pas complété les questionnaires . . . . .	104

E	Population qui a participé à l'étude	106
F	Réponses des étudiants au premier questionnaire d'auto-évaluation	108
G	Réponses des étudiants au questionnaire de mathématiques	109
H	Réponses des étudiants au questionnaire de programmation en langage Python	111
I	Réponses des étudiants au questionnaire post-examen	113

## Acronymes

**CAFE** Correction Automatique et un Feedback aux Etudiants. 11.

**CECRL** Cadre Européen Commun de Référence pour les Langues. 23, 33, 108.

**CM** Certitude Moyenne. 16, 17, 38, 39, 41, 43.

**DC** Degré de Certitude. 17, 18, 38, 42, 47, 48, 109, 110, 111, 112.

**EC** Errer de Centration. 16, 38, 39, 41, 43.

**GRIEMetic** Groupe de Recherche Interuniversitaire sur l'Evaluation et la Mesure en Education à l'aide des tic. 20.

**IDE** Integrated Development Environment. 30.

**IHM** Interaction Humain Machine. 19.

**POGIL** Process Oriented Guided Inquiry Learning. 13, 57.

**PPIG** Psychology of Programming Interest Group. 18.

**QCM** Questionnaire à Choix Multiples. 24, 25, 29, 30.

**RGPD** Règlement Général sur la Protection des Données. 22.

**TE** Taux d'Exactitude. 16, 17, 38, 39, 41, 43.



# 1 Introduction

L'apprentissage de la programmation est un élément clé dans les études en sciences informatiques, en particulier en première année. La programmation est une compétence fondamentale pour les étudiants car cela leur permet de comprendre les concepts basiques de l'informatique et d'acquérir de solides bases pour la suite de leur cursus et des projets futurs. Cependant, les taux d'abandon y sont souvent élevés ; ce qui peut être lié en partie, pour certains étudiants, à la difficulté de cet apprentissage.

De nombreuses recherches ont été effectuées pour comprendre ces abandons ainsi que les échecs au cours d'introduction à la programmation. Ces multiples recherches ont parcouru de nombreux facteurs de réussite, d'outils mis en place et de méthodes d'enseignements afin d'élaborer des modèles pour prédire et identifier précocement les étudiants qui risquent d'échouer.

L'auto-évaluation est un de ces facteurs et il permet notamment à un étudiant de se positionner par rapport à ses connaissances, que ce soit sur ses capacités ou sur l'appropriation de la matière académique. Le but de notre recherche est donc d'explorer l'impact du facteur d'auto-évaluation au travers de questionnaires techniques en programmation et en mathématiques afin d'aider les étudiants à se positionner par rapport à leurs acquis.

Notre problématique est établie en deux parties. Premièrement, nous nous penchons sur les faiblesses qui peuvent être identifiées par l'utilisation de l'auto-évaluation. Pour ce faire, nous décidons de réaliser quatre questionnaires que nous soumettons aux étudiants inscrits au cours *INFOB131 - Introduction à la programmation* à l'Université de Namur début décembre, environ un mois précédant les examens.

Le premier questionnaire consiste en une évaluation, par les étudiants, de leurs connaissances en mathématiques et en programmation en langage Python.

Ensuite, nous soumettons deux questionnaires techniques sur ces matières en demandant aux étudiants de préciser, pour chaque réponse, quel est, selon eux, le degré de certitude avec lequel ils ont répondu.

Début février, nous soumettons un dernier questionnaire aux étudiants afin d'avoir un retour sur plusieurs points :

- leur participation ou non aux questionnaires soumis
- leur réussite ou non à l'examen du cours d'introduction à la programmation
- en cas de non participation aux précédents questionnaires, la raison pour laquelle ils ne les ont pas complétés
- en cas de participation aux précédents questionnaires, ce qu'ils aimeraient/souhaiteraient

voir dans un module d'aide à la réussite

L'analyse des données recueillies via les différents questionnaires permettrait de mieux comprendre l'impact de l'auto-évaluation sur la réussite du cours d'introduction à la programmation en première année; que ce soit par rapport aux matières ou par rapport à la façon dont les étudiants évaluent leurs connaissances.

La deuxième partie de notre problématique se concentre sur des pistes de recherches futures pour approfondir notre compréhension de l'impact de l'auto-évaluation sur l'apprentissage de la programmation en première année en informatique, ainsi que sur des outils à destination des enseignants pour aider les étudiants à développer leurs compétences.

Pour la suite de ce mémoire, nous présentons une revue de la littérature effectuée sur des études portant sur différents facteurs de réussite pour un cours d'introduction à la programmation, sur le facteur d'auto-évaluation, sur les matières visées par les questionnaires de notre recherche et enfin sur la centration et les degrés de certitude.

Suite à la confection de l'état de l'art, nous détaillons notre problématique. Ensuite, nous expliquons la méthodologie que nous avons appliquée pour réaliser notre recherche. Nous examinons la population visée par l'étude, la création des questionnaires, la passation de ceux-ci et la collecte des résultats.

Après la méthodologie, nous présentons les résultats collectés. L'interprétation de ceux-ci est ensuite effectuée dans la partie discussion.

Enfin, nous concluons notre travail.

## 2 État de l’art

### 2.1 Contexte général

Le taux de réussite des étudiants universitaires de première année au cours d’introduction à la programmation n’est pas élevé.

En 2007, dans [1], J. Bennedsen et M. Caspersen estiment qu’il est problématique de ne pas disposer de preuves solides par rapport au faible taux de réussite à des cours d’introduction à la programmation. Ils ont donc décidé de mener une étude afin d’obtenir un taux de réussite au niveau mondial. Pour ce faire, ils ont mené une enquête auprès de 575 institutions de par le monde proposant un programme en informatique. Bien que seulement 63 retours aient pu être pris en considération, ils ont établi que le taux de réussite moyen s’élevait à 67%.

Dans [2], C. Watson et F. Li décident, en 2014, d’apporter des preuves substantielles supplémentaires sur ce taux de réussite. Ils effectuent un examen systématique de la littérature sur les cours d’introduction à la programmation et examinent donc les taux de réussite de 161 cours venant de 51 institutions réparties dans 15 pays différents. Tout comme les auteurs de [1], ils décident de déterminer le taux de réussite mais aussi d’examiner comment les taux de réussite varient dans le temps. En ce qui concerne le taux de réussite, ils arrivent à une conclusion similaire que [1] avec un taux de réussite de 67,7%. Enfin, ils concluent que le taux de réussite n’a pas varié de manière significative au cours du temps, comme le montre la figure 1.

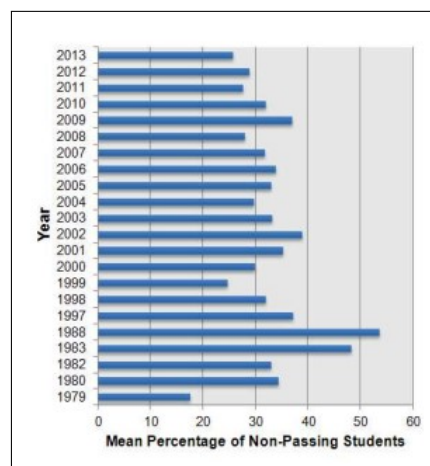


FIGURE 1 – Pourcentages moyens d’étudiants n’ayant pas réussi en fonction de l’année au cours de laquelle le cours a été dispensé tels que présentés dans [2].

En 2019, J. Bennedsen et M. Caspersen ont décidé de réitérer leur étude de 2007. Dans [3], ils ont obtenu 161 retours exploitables sur un total de 998 demandes envoyées aux différents établissements. Dans cette recherche, le taux de réussite moyen obtenu a sensiblement augmenté passant de 67% en 2007 à 72% en 2019. Néanmoins, les auteurs concluent que l’initiation des

étudiants à l'informatique reste l'un des grands défis de l'enseignement de l'informatique.

De façon plus locale, nous pouvons également mettre en avant l'étude réalisée par C. Libert et W. Vanhoof dans [4] en 2018 dans laquelle ils relèvent que le taux de réussite en première session du cours de programmation d'étudiants de première année en sciences mathématiques n'est pas très élevé et est d'environ 42%.

Afin d'expliquer les raisons du taux de réussite pas très élevé des étudiants universitaires de première année au cours d'introduction à la programmation, de prédire les résultats de ces étudiants ou de dégager des solutions pour pallier ce problème d'échecs, de nombreuses recherches ont été menées.

Nous avons également entrepris une revue de la littérature afin d'appréhender des initiatives existantes visant à soutenir les étudiants dans leurs cursus universitaire.

Afin de structurer notre analyse, nous présenterons dans les sous-sections suivantes les différentes approches pour accompagner les étudiants ainsi que les dispositifs mis en place pour favoriser leur réussite académique.

### **2.1.1 Identification précoce des étudiants à risque**

Certaines recherches ont porté sur l'identification précoce des étudiants en difficulté afin de leur fournir un soutien adapté.

Une partie de ces études se basent sur l'analyse de données récoltées au moyen d'outils et à la conception de modèles permettant de prédire l'échec d'un étudiant. Dans [5], les auteurs analysent l'interaction des étudiants avec des outils pédagogiques basés sur le web ; tout comme dans [6] où les modèles sont élaborés sur base de questions soumises en classe. D'autres études se sont basées sur des outils de capture de codes comme [7] où les chercheurs ont analysé les transcriptions détaillées des frappes de clavier et les résultats des tentatives de compilation pendant les activités de programmation ou dans [8] où les chercheurs ont utilisé un outil capable de capturer de manière autonome et discrète des informations en temps réel sur les performances des élèves en se basant sur des instantanés de leur code.

Une autre partie des recherches se basent quant à elles sur la collecte de données telles que des catégories de préparation et d'attitude [9] ou les notes aux devoirs et premier test [10] ; mais aussi des données relatives aux tâches d'évaluation formative des deux premières semaines du semestre qui ont permis aux chercheurs de [11] d'élaborer un modèle de prédiction par arbre de classification. Dans [12], les auteurs collectent les données d'environnements web et après traitement les mettent en corrélation avec les notes des étudiants.

Enfin, dans [13], les auteurs suggèrent, à l'aide d'un outil d'apprentissage (BitFit), que les

étudiants à risque peuvent être identifiés dès les deux premières semaines du semestre .

Les auteurs de [4] de l'Université de Namur ont souligné l'importance donnée à la fonction de régulation pour améliorer l'apprentissage des étudiants. Leur étude s'est concentrée sur une année académique spécifique où un processus d'évaluation continue a été mis en place pour un cours d'introduction à la programmation.

Les résultats de cette étude sont basés sur une demi-année. Bien que ceux-ci soient encourageants, les auteurs précisent que la continuation de ce dispositif dépendra des résultats obtenus en fin d'année.

### **2.1.2 Utilisation d'outils numériques**

Dans un second temps, des études se sont concentrées sur l'utilisation d'outils numériques pour aider les étudiants dans leur apprentissage mais également pour aider les professeurs à identifier précocement les problèmes éventuels afin d'y donner une réponse rapide.

Dans [14], les auteurs se sont concentrés sur l'utilisation de l'outil d'apprentissage en ligne Canvas et plus précisément sur les conversations entre les étudiants au sujet de difficultés à appréhender la matière, etc. Dans [15], les chercheurs présentent un ensemble de stratégies, incluses dans une application, dont l'objectif est d'aider les enseignants et les étudiants. Enfin, dans [16], il est fait état de l'étude de l'utilisation par les étudiants de matériel d'apprentissage en ligne et que l'information peut être utilisée pour identifier les étudiants à risque.

Dans [17], les auteurs ont mis en place une plateforme en ligne appelée "CAFE" qui permet une Correction Automatique et un Feedback aux Etudiants de manière précise pour maximiser les chances de l'étudiant à s'auto-réguler. Cette plateforme permet une analyse des résultats du programme soumis par l'étudiant et une analyse sur le processus cognitif à la construction du programme.

Il est également intéressant de citer INGIInious<sup>1</sup>. Cet outil a été développé par le département d'ingénierie informatique de l'Université Catholique de Louvain. C'est une plateforme informatique interactive d'aide pour l'apprentissage de la programmation. Un étudiant soumet un programme pour répondre à des exercices proposés par les enseignants. INGIInious exécute le programme et l'étudiant a un retour immédiat. Les enseignants ont également la possibilité d'effectuer un suivi des étudiants et de personnaliser un feedback complémentaire.

### **2.1.3 Performance des étudiants**

Un autre facteur qui a fait l'objet de différentes recherches est la performance des étudiants.

---

1. <https://ii1.su/YbGHt> et <https://ii1.su/T8ejz>

Dans [18], les chercheurs, afin de prédire le plus précocement possible la note finale, ont mesuré le degré de corrélation entre les performances des étudiants lors de quizz, de laboratoire, de programmes et de tests par rapport à la note finale du cours. Ils ont conclu qu'il y a une forte corrélation positive pour les quatre types d'évaluation.

Une autre étude s'est plus centrée sur les étudiants. Dans [19], les auteurs se sont penchés sur l'analyse des performances des étudiants et leur amélioration dans les langages de programmation. L'enquête générale et les tests (de raisonnement, de logique et sur les concepts techniques) ont été réalisés à l'aide de formulaires Google. Le modèle élaboré sur base des résultats permet d'identifier les étudiants qui apprennent moins rapidement et permet aux étudiants d'améliorer leurs concepts les plus faibles.

Pour les auteurs de [20], utiliser une prédiction automatisée des performances des étudiants à un stade précoce est utile pour l'apprentissage. En effet, en identifiant les étudiants susceptibles d'échouer à un stade initial, des mesures correctives peuvent être prises pour améliorer leurs résultats. Ici, l'apprentissage automatique utilisé est la technique de regroupement récursif qui a pour but de regrouper les étudiants suivant leurs performances au cours et de prendre en considération les étudiants des groupes inférieurs car ils sont susceptibles d'échouer.

Par l'approche IPSSDD (détection et développement itératif de stratégies de résolution de problèmes), les chercheurs de [21] impliquent directement les étudiants dans tous les aspects de la résolution de problèmes, y compris la détection, le développement, l'affinement et l'application de stratégies de résolution de problèmes. En commençant par les bases, les instructions if et les boucles, les auteurs ont constaté que c'est dans les boucles que de nombreux étudiants échouent. Ils ont également remarqué que les sections qui utilisaient l'approche IPSSDD obtenaient de meilleurs résultats et que les étudiants de ces sections avaient également une plus grande motivation ainsi qu'une plus grande confiance en eux.

#### **2.1.4 Passeport pour le bac**

En Belgique, l'Université de Namur, l'Université de Saint-Louis Bruxelles et l'Université Catholique de Louvain ont lancé le "Passeport pour le Bac" afin de permettre aux étudiants entrant à l'université de mieux se préparer en travaillant sur leurs capacités à s'auto-réguler pour développer leurs compétences à s'auto-évaluer.

*"Le "Passeport pour le Bac" est une aide à la transition entre l'enseignement secondaire et l'université et a pour vocation d'avoir :*

- *des visées d'action : promouvoir et démocratiser la réussite des étudiants de 1ère année universitaire par une triple intervention :*

- *Identification des prérequis des formations*
  - *Mesure de leur maîtrise : les Passeports*
  - *Mise en place d'actions correctrices*
- *des visées de recherche :*
    - *Identification des prérequis des formations*
    - *Mieux comprendre les mécanismes par lesquels ils opèrent*
    - *Établir des relations entre leur maîtrise et les données sociobiographiques et scolaires*
    - *Observatoire des acquis à l'entrée de l'université (à plus long terme)"<sup>2</sup>*

Diverses études en lien avec le "Passeport pour le Bac" ont donc été réalisées [22], [23], [24].

Dans [23], les auteurs ont réalisé une recherche sur le développement de l'autonomie des étudiants.

Cette étude vise à explorer l'estimation de la maîtrise des prérequis en début d'année afin de mieux comprendre son impact sur les performances académiques des étudiants. L'accent est mis sur l'apprentissage actif et l'engagement des étudiants dans leur propre progression, tout en intégrant des évaluations formatives pour suivre l'évolution dans leur apprentissage. De plus, cette étude met en place l'utilisation d'un degré de certitude basé sur 3 échelles (5/5 (bonne maîtrise), 4/5 (partiellement acquis) et < 4/5 (non acquis)) pour examiner l'écart entre les compétences perçues par les étudiants et leur évaluation réelle pour une matière donnée.

## 2.2 L'auto-efficacité

D'autres chercheurs se sont focalisés sur le facteur d'auto-efficacité.

Dans [25], les auteurs définissent l'auto-efficacité comme étant le jugement personnel d'une personne sur sa capacité à accomplir une tâche et regardent comment une restructuration des devoirs peut avoir un impact sur l'auto-efficacité et la performance des étudiants.

Une autre approche est celle de l'apprentissage collaboratif POGIL afin de développer certaines "compétences de processus" de l'étudiant telles que la communication, la pensée critique ou le travail d'équipe. Dans [26], les auteurs examinent les perceptions des étudiants sur leur apprentissage du contenu et sur leurs compétences en matière de processus et explorent la relation entre l'auto-efficacité, les performances d'apprentissage et ces perceptions. Ils concluent que seule l'auto-efficacité est un facteur prédictif significatif de l'apprentissage des étudiants.

Les recherches menées dans [27] sont intéressantes dans le sens où cette étude présente les

---

2. Source : <https://ii1.su/4QFX1>

résultats d'un essai d'intervention visant à améliorer l'auto-efficacité des étudiants dans le cadre d'un cours d'introduction à la programmation. Il a été demandé à certains étudiants de réaliser une tâche d'auto-évaluation. Les chercheurs ont constaté que ces étudiants ont obtenu des résultats plus élevés que ceux qui n'ont pas réalisé cette tâche d'auto-évaluation; même s'il y a peu de preuves d'un effet sur l'auto-efficacité par l'auto-évaluation. En conclusion, les auteurs suggèrent d'intégrer des tâches d'auto-évaluation dans les cours d'introduction à la programmation.

## 2.3 L'auto-évaluation

L'objectif de notre recherche est d'étudier le facteur d'auto-évaluation (formative) sur l'apprentissage de la programmation.

Comme nous l'avons relevé plus haut, les auteurs de [27] suggèrent d'intégrer des tâches d'auto-évaluation dans les cours d'introduction à la programmation.

Le facteur d'auto-évaluation est un sujet de recherche important dans le domaine de l'enseignement de la programmation en première année de science informatique. En effet, l'auto-évaluation est un outil qui permet aux étudiants de prendre conscience de leurs propres compétences et de leur niveau de confiance en eux, ce qui peut avoir un impact sur leur motivation et leur réussite académique. Dans ce contexte, plusieurs travaux de recherche ont été menés pour évaluer l'efficacité de l'auto-évaluation dans l'enseignement de la programmation.

Dans [28] les auteurs ont conclu que l'auto-évaluation peut être bénéfique pour les étudiants en leur permettant de mieux comprendre leur propre apprentissage, mais qu'elle nécessite une orientation et un soutien appropriés pour être efficace.

Il a été relevé, dans [29], que les étudiants ayant utilisé Quiz-PACK, une plateforme proposant des exercices d'auto-évaluation personnalisés pour les étudiants en programmation, ont obtenu de meilleurs résultats à un examen final et les auteurs soulignent également que QuizPACK permet aux étudiants de s'auto-évaluer et de comprendre leurs propres forces et faiblesses, ce qui peut les aider à mieux cibler leur apprentissage.

De plus, dans [30], les auteurs ont constaté que la participation fréquente à des quiz d'auto-évaluation est corrélée à une meilleure performance aux examens. Afin d'établir des questionnaires comprenant de l'auto-évaluation, il est judicieux de se référer à [31] et à [32] où l'auteur explicite l'utilisation des questionnaires avec degrés de certitude et le calcul de l'indice de centration permettant de mesurer la manière dont l'étudiant s'auto-évalue.



## 2.4 Auto-efficacité et auto-évaluation, deux facteurs identiques ?

Les auteurs de [25], [26] et [27] s'appuient tous sur [33] pour définir le facteur d'auto-efficacité comme étant "le jugement personnel d'une personne sur sa capacité à accomplir un objectif ou une tâche". Dans [25], les auteurs précisent également que l'auto-efficacité peut affecter le niveau d'effort et la capacité d'un élève à persévérer.

Pour l'auto-évaluation, [30], les auteurs notent que ce facteur constitue une compétence métacognitive essentielle qui permet aux étudiants de gérer plus efficacement leur temps et leurs ressources pour l'apprentissage. Si un étudiant repère des lacunes dans sa compréhension, il peut de manière stratégique répartir son temps et son énergie vers des sujets plus prioritaires.

Comme mentionné dans la section 2.2, dans [27], des étudiants ont été sélectionnés pour effectuer une tâche d'auto-évaluation qui pourrait influencer positivement leur auto-efficacité, et par conséquent, potentiellement améliorer leurs performances académiques. Les conclusions de cette étude montrent que bien qu'il y ait peu de preuves directes d'une relation entre l'auto-évaluation et l'auto-efficacité, les étudiants ayant accompli la tâche d'auto-évaluation ont obtenu des scores plus élevés par rapport à ceux n'ayant pas participé à cette tâche.

L'auto-efficacité et l'auto-évaluation présentent donc des similitudes tout en étant distinctes et peuvent interagir mutuellement.

## 2.5 Matières visées par la recherche

Pour effectuer notre recherche, nous comptons donc faire passer des questionnaires aux étudiants, questionnaires portant sur le langage de programmation Python mais également sur les mathématiques.

En effet, plusieurs recherches ont été effectuées en utilisant du code écrit par les étudiants ([34], [35]) ou en proposant des blocs de codes en demandant aux étudiants de trouver la sortie ou l'entrée correspondantes à l'énoncé ([12], [36]).

Enfin, la connaissance des mathématiques est importante dans la formation informatique. [37] et [38] soulignent que l'apprentissage des mathématiques est une partie essentielle de la formation en informatique, car il permet aux étudiants de comprendre les concepts fondamentaux et d'acquérir les compétences nécessaires pour résoudre des problèmes algorithmiques de manière efficace et qu'ils doivent s'efforcer d'acquérir des compétences solides en mathématiques pour réussir dans ce domaine.

## 2.6 Centration et degrés de certitude

L'utilisation des degrés de certitude dans les questionnaires à choix multiple est détaillée par Gilles dans [31] et [32].

Il est demandé aux étudiants de se positionner suivant une échelle de probabilité (voir figure 2) où chaque degré de certitude représente un intervalle de pourcentage de certitude.

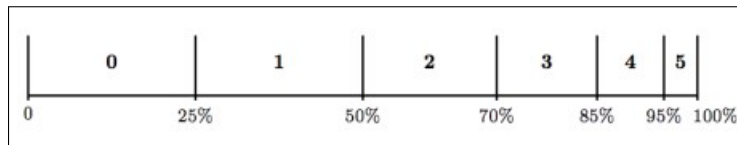


FIGURE 2 – Echelle de probabilité des degrés de certitude

Le degré de certitude choisi par l'étudiant conditionnera sa note suivant l'échelle des degrés de certitude (reprise à la figure 3).

		Degrés de certitude					
		0	1	2	3	4	5
Réponse	Correcte	+13	+16	+17	+18	+19	+20
	Incorrecte	+4	+3	+2	0	-6	-20

FIGURE 3 – Echelle de score des degrés de certitude

Enfin, afin de déterminer si l'étudiant se surestime ou se sous-estime, l'indice de centration doit être calculé :

### *"Mesure de l'Erreur de Centration*

*Le calcul de l'indice de centration est basé sur la différence entre la certitude moyenne (CM) et le taux d'exactitude (TE). Les scores des Erreurs de Centration (EC) montrent en fonction du signe la tendance à se surestimer (+) ou à se sous-estimer (-). L'étendue des valeurs possibles va de -100 à +100. Une performance idéale vaut 0."*<sup>3</sup>

$$EC = CM - TE$$

Pour calculer la certitude moyenne de l'étudiant, nous nous basons sur l'échelle de probabilité (figure 2) et nous prenons les valeurs centrales des intervalles des degrés de certitude (voir tableau1).

La certitude moyenne de l'étudiant se calcule comme la somme du nombre d'utilisation totale

---

3. J-L. GILLES dans [32]

Degrés de certitude	Valeurs centrales
DC0	0.125
DC1	0.375
DC2	0.6
DC3	0.775
DC4	0.9
DC5	0.975

TABLE 1 – Valeurs centrales des intervalles des degrés de certitude

de chaque degré de certitude multiplié par la valeur centrale correspondante, le tout divisé par le nombre total de questions et multiplié par 100 :

$$CM = (\sum_{i=0}^5 VC_i * n_i) * \frac{100}{q},$$

où  $i$  représente le degré de certitude,  $VC_i$  la valeur centrale correspondante,  $n_i$  le nombre d'utilisation du degré de certitude  $i$  et  $q$ , le nombre total de questions.

Le taux d'exactitude se calcule comme la somme du nombre total de réponses correctes divisé par le nombre total de questions et multiplié par 100 :

$$TE = (\sum_{i=0}^5 m_i) * \frac{100}{q},$$

où  $i$  représente le degré de certitude,  $m_i$  le nombre de réponses correctes en utilisant le degré de certitude  $i$  et  $q$ , le nombre total de questions.

Ensuite, en comparant l'erreur de centration de l'étudiant à l'échelle de centration (figure 4), nous pouvons déterminer si celui-ci se surestime ou se sous-estime.

<b>L'INDICE DE CENTRATION REFLETE :</b>	
<b><u>UNE AUTO-ESTIMATION</u></b>	
<b>EXCELLENTE</b>	si elle est comprise entre <b>-4 et +4</b>
(avec surestimation très modérée si signe positif, ou sous-estimation très modérée si signe négatif)	
<b><u>UNE SOUS-ESTIMATION</u></b>	
<b>TRES LEGERE</b>	si elle est comprise entre <b>-5 et -9</b>
<b>LEGERE</b>	<b>-10 et -14</b>
<b>ASSEZ FORTE</b>	<b>-15 et -19</b>
<b>FORTE</b>	<b>-20 et -24</b>
<b>TRES FORTE</b>	<b>-25 et -29</b>
<b>EXTREME</b>	<b>-30 et -100</b>
<b><u>UNE SURESTIMATION</u></b>	
<b>TRES LEGERE</b>	si elle est comprise entre <b>+5 et +9</b>
<b>LEGERE</b>	<b>+10 et +14</b>
<b>ASSEZ FORTE</b>	<b>+15 et +19</b>
<b>FORTE</b>	<b>+20 et +24</b>
<b>TRES FORTE</b>	<b>+25 et +29</b>
<b>EXTREME</b>	<b>+30 et +100</b>

FIGURE 4 – Echelle de centration définie par Gilles

Nous recourons également aux catégories du vecteur des degrés de connaissance de Leclercq & Plunus [39] (table 2) :

Réponses correctes			Réponses incorrectes		
DC0-DC1	DC2-DC3	DC4-DC5	DC0-DC1	DC2-DC3	DC4-DC5
Compétence ignorée	Compétence partielle	Compétence assurée	Ignorance reconnue	Ignorance partielle	Ignorance ignorée

TABLE 2 – Catégories du vecteur des degrés de connaissances

## 2.7 La psychologie de la programmation

Dans [40], J. Sajaniemo définit la psychologie de la programmation comme un domaine interdisciplinaire qui englobe :

- la recherche sur la cognition des programmeurs informatiques ;
- les méthodes et les outils en lien avec les activités relatives à la programmation ;
- l'enseignement de la programmation.

Toujours dans [40], l'auteur retrace certains événements de l'atelier du PPIG de juillet 2007 en Finlande et revient sur 5 articles sélectionnés parmi ceux présentés lors de cet atelier et publiés dans un numéro spécial de *Human Tehnology*<sup>4</sup>.

4. Voir <https://ht.csr-pub.eu/index.php/ht/index>

Un de ceux-ci, [41] de S. Jones et G. Burnett, se penche sur le problème beaucoup étudié qu'est la prédiction de la réussite des étudiants dans l'apprentissage de la programmation. Dans leur étude, les auteurs étudient la capacité spatiale et son impact sur l'apprentissage de la programmation et concluent qu'elle est importante dans cet apprentissage. Cependant, ils jugent qu'elle ne doit pas être utilisée comme moyen de déterminer l'aptitude à la programmation mais plutôt être prise en compte pour la conception d'interventions pédagogiques.

J-M. Hoc, quant à lui, souligne, dans [42], que les travaux des recherches en psychologie de la programmation dans la langue française peuvent être regroupés sur trois grands thèmes :

- L'analyse du travail du programmeur.
- La formation du programmeur (chez des sujets non informaticiens et chez des étudiants en programmation).
- Travaux d'évaluation (évaluer diverses caractéristiques de l'environnement de programmation).

Enfin, dans [43], les auteurs passent en revue 50 années sur l'évolution de la psychologie de la programmation. Ils soulignent le fait que la psychologie de la programmation est un domaine particulier de l'IHM mais qu'il existe de nombreuses communautés pour lesquelles la psychologie de la programmation peut intervenir et où des recherches sont publiées.

La figure 5 reprend certaines de ces communautés et domaines dans lesquels de la littérature sur la psychologie de la programmation peut être trouvée.

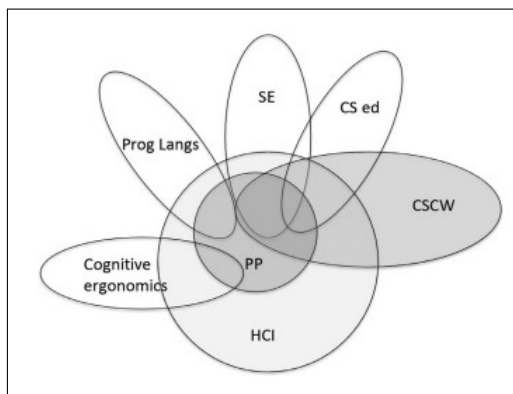


FIGURE 5 – Communautés de recherche recoupant la psychologie de la programmation (PP) : ergonomie cognitive, langages de programmation, génie logiciel (SE), enseignement de l'informatique (CS ed), travail coopératif assisté par ordinateur (CSCW), interaction homme-machine (HCI). Figure et description reprises de [43].

### 3 Problématique

Dans cette section, nous positionnons notre étude et présentons les questions de recherche que nous avons définies afin de déterminer l'incidence du facteur d'auto-évaluation dans l'apprentissage de la programmation ainsi que les hypothèses que nous émettons par rapport à celles-ci.

#### 3.1 Positionnement de la recherche

Dans l'article du 18 décembre 2020 publié sur le site du GRIEMetic, les professeures de l'Université du Québec à Montréal D.T.D. Thi et C. B. Da Costa signalent que "*lorsque l'étudiant s'autoévalue, il prend conscience non seulement de ses processus cognitifs, mais également de ses motivations et de son sentiment d'efficacité à effectuer la tâche proposée*"<sup>5</sup>.

En s'appuyant sur cette citation et sur la définition de la psychologie de la programmation donnée par J. Sajaniemo dans [40] (voir section 2.7) et en se référant aussi sur [43] précisant que la psychologie de la programmation peut intervenir dans de nombreux domaines et communautés, nous situons notre travail à l'intersection des domaines de la psychologie de la programmation et de l'enseignement de l'informatique.

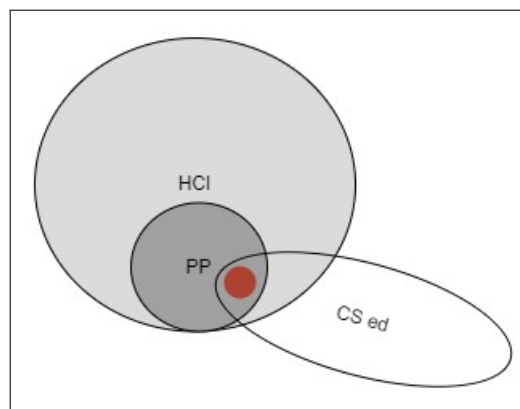


FIGURE 6 – Positionnement de notre mémoire par rapport aux domaines de recherche de l'interaction humain-machine (HCI), de l'enseignement de l'informatique (CS ed) et de la psychologie de la programmation (PP)

#### 3.2 Questions de recherche

Notre étude est centrée sur deux thèmes principaux :

- l'identification des faiblesses des étudiants
- l'amélioration de la détection de problèmes d'apprentissage en utilisant le facteur d'auto-évaluation

---

5. Voir <https://ii1.su/8U1Gy>

En effet, notre objectif principal est de comprendre comment l'auto-évaluation peut être utilisée pour identifier les faiblesses des étudiants et comment elle peut être utilisée pour aider les enseignants à cibler des lacunes ou des problèmes d'apprentissage mais aussi à permettre aux étudiants de se situer par rapport à la matière.

Nos questions de recherche découlent naturellement de notre objectif principal et nous les détaillons ici.

### **3.2.1 QR1 : Quelles faiblesses dans l'apprentissage de la programmation l'auto-évaluation permet-elle d'identifier ?**

Nous émettons l'hypothèse que l'utilisation du facteur d'auto-évaluation dans le cadre d'un cours d'introduction à la programmation permet d'identifier des faiblesses auprès des étudiants telles que des lacunes dans la connaissance de la matière mais aussi des problèmes dans l'appropriation de cette matière ou encore des difficultés à résoudre des problèmes.

Deuxièmement, nous pensons que le facteur d'auto-évaluation permet également de détecter des problèmes dans la manière dont les étudiants estiment leur niveau de connaissance. En se référant aux études de Gilles [31] et [32] et en appliquant le calcul de l'indice de centration, nous souhaitons examiner si les étudiants ont tendance à sous-estimer ou surestimer leurs connaissances.

Enfin, nous souhaitons également savoir si l'utilisation du facteur d'auto-évaluation fournit une aide aux étudiants pour la réussite du cours d'introduction à la programmation.

### **3.2.2 QR2 : Quelles améliorations peuvent être envisagées suite à l'intégration de l'auto-évaluation dans l'apprentissage de la programmation ?**

Par cette deuxième question de recherche, nous souhaitons explorer différentes possibilités pour améliorer l'utilisation de l'auto-évaluation :

- Le développement d'un outil ou de modules d'apprentissages
- Une aide à destination des professeurs ou des assistants
- Un feedback vers les étudiants

Ces améliorations seront détaillées dans la section Discussion.

## 4 Méthodologie

Dans cette section, nous examinons la méthodologie appliquée pour réaliser notre recherche.

Nous décidons de faire passer différents questionnaires aux étudiants du cours de jour d'introduction à la programmation en deux étapes.

Dans un premier temps, afin d'avoir une vue d'ensemble des connaissances générales des étudiants et savoir comment l'étudiant s'auto-évalue, nous réalisons quatre questionnaires que nous soumettons aux étudiants début décembre.

Pour la deuxième étape, nous réalisons un questionnaire post-examen que nous soumettons à l'ensemble des étudiants début février, même s'ils n'ont pas complété les questionnaires précédents.

Nous allons donc expliciter la population visée, la création des questionnaires, la passation de ceux-ci et la collecte des résultats.

### 4.1 Population visée

Pour la réalisation de notre étude, nous avons sollicité les étudiants inscrits au cours *INFOB131 - Introduction à la programmation* dispensé par le professeur Benoît FRENAY. Ceci regroupe donc les étudiants de 1ère bachelier en sciences informatiques mais aussi les étudiants de 1ère bachelier en sciences économiques (orientation ingénieur de gestion).

Nous avons listé l'ensemble des étudiants qui ont participé, à divers degrés, à notre recherche. Comme nous avons anonymisé les résultats, nous avons attribué des identifiants aux étudiants. Le tableau 20 à l'annexe E reprend ces données.

### 4.2 Création des questionnaires

Afin de se conformer à la législation sur la protection des données (RGPD), l'ensemble des résultats des différents questionnaires ont été anonymisés.

La création des questionnaires se fait en deux temps. Premièrement nous avons réalisé quatre questionnaires que nous avons soumis aux étudiants début décembre, avant de passer l'examen relatif au cours.

Dans un second temps, nous avons rédigé un questionnaire post-examen qui a été soumis aux étudiants début février.

Afin de permettre à un maximum d'étudiants d'avoir un accès aisé aux questionnaires, nous



décidons d'utiliser l'outil Google Forms.

#### 4.2.1 Création des premiers questionnaires

Les quatre questionnaires établis sont les suivants :

- Le premier questionnaire est une demande d'auto-évaluation de la part de l'étudiant quant à ses connaissances en mathématiques, son niveau d'anglais, ses connaissances en langage de programmation python et s'il connaît d'autres langages de programmation.
- Le second questionnaire reprend des exercices de mathématiques suivant la matière dispensée dans le cadre du cours *INFOB125 - Fondements mathématiques pour l'informatique (1re partie)*.
- Le troisième questionnaire reprend des exercices relatifs au langage de programmation python suivant la matière dispensée dans le cadre du cours *INFOB131 - Introduction à la programmation*. Nous nous sommes également appuyés sur [44] afin de réaliser ce questionnaire.
- Le dernier questionnaire est identique au premier questionnaire et permet à l'étudiant d'ajuster son auto-évaluation suite au passage des deux questionnaires techniques.

##### Premier questionnaire

Nous demandons à l'étudiant de positionner ses compétences sur une échelle de 0 à 10 sur les matières suivantes :

- Algèbre
- Mathématiques discrètes
- En programmation en utilisant le langage Python

Au niveau de l'algèbre et des mathématiques discrètes, nous incluons des exemples afin de permettre aux étudiants de bien cibler la matière sur laquelle nous l'interrogeons (un exemple est repris à la figure 7).

Pour la connaissance de la langue anglaise, nous demandons à l'étudiant de communiquer son niveau suivant la norme CECRL qui comporte les niveaux suivants :

- **A0** : Aucune notion
- **A1** : Débutant - peut comprendre et utiliser des expressions familières
- **A2** : Peut communiquer lors de tâches simples et habituelles ne demandant qu'un échange d'informations simple et direct sur des sujets familiers et habituels

A combien situerais-tu tes connaissances en Algèbre ? \*

Pour rappel, l'algèbre reprend notamment :

- algèbre des matrices
- déterminant

<p style="text-align: center;"><b>Algèbre des matrices</b></p> $3 \begin{bmatrix} 3 & -1 & \sqrt{2} \\ \sqrt{3} & 1 & 0 \\ 2 & 3 & -1 \end{bmatrix} - 2 \begin{bmatrix} 4 & -1 & 3\sqrt{2} \\ 2\sqrt{3} & 0 & 1 \\ 3 & -1 & 2 \end{bmatrix}$	<p style="text-align: center;"><b>Déterminant d'une matrice</b></p> <p>Soit la matrice</p> $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ <p>le déterminant est</p> $\det A = a_{11}(-1)^{1+1} \det a_{22} + a_{12}(-1)^{1+2} \det a_{21}$
--	--

0 1 2 3 4 5 6 7 8 9 10

Aucune connaissance            Parfaite connaissance

FIGURE 7 – Exemple de question pour le premier questionnaire

- **B1** : Intermédiaire - peut se débrouiller dans la plupart des situations rencontrées en voyage dans une région où la langue cible est parlée
- **B2** : Courant - peut communiquer avec un degré de spontanéité et d'aisance tel qu'une conversation avec un locuteur natif
- **C1** : Avancé - peut s'exprimer sur des sujets complexes de façon claire et bien structurée
- **C2** : Bilingue - peut s'exprimer spontanément, très couramment et de façon précise. Peut rendre distinctes de fines nuances de sens en rapport avec des sujets complexes

Enfin, nous interrogeons l'étudiant sur le(s) éventuel(s) langage(s) de programmation qu'il connaît (réponse ouverte).

**L'ensemble du questionnaire se trouve à l'annexe A.**

### Questionnaire d'exercices de mathématiques

Dans ce questionnaire, nous parcourons la matière du cours *INFOB125 - Fondements mathématiques pour l'informatique (1re partie)* sous forme de QCM en demandant aux étudiants, pour chaque question, de choisir un degré de certitude allant de 0 à 5.

Le questionnaire comporte 27 questions réparties comme suit :

- **Algèbre**
  - Algèbre des matrices : 4 questions

- Calcul du déterminant de matrice : 1 question

- **Mathématiques discrètes**

- Logique des propositions : 5 questions

- Logique des prédicats : 4 questions

- Calcul booléen : 4 questions

- Système de numération et de représentation : 5 questions

- Codage-décodage : 4 questions

Après concertation avec notre co-promotrice, il a été convenu de ne pas donner de cotation aux étudiants ; le but étant, ici, qu'ils puissent s'auto-évaluer et non pas de passer un pré-examen.

**L'ensemble du questionnaire se trouve à l'annexe B.**

Questionnaire d'exercices de python

Dans ce questionnaire, nous parcourons la matière du cours *INFOB131 - Introduction à la programmation* sous forme de questions ouvertes (donner la sortie à l'exécution d'une partie de code, réaliser du code sur base d'un énoncé) ou de questions de type QCM.

Ici aussi, nous demandons aux étudiants de choisir un degré de certitude allant de 0 à 5 pour chaque question.

L'ensemble du questionnaire a été validé par un assistant du cours *INFOB131 - Introduction à la programmation*.

Le questionnaire comporte 23 questions réparties comme suit :

- **Les variables** : 2 questions
- **Les conditions** : 3 questions
- **Les fonctions** : 3 questions
- **Les boucles** : 4 questions
- **Les structures de données** : 7 questions
- **Les fichiers** : 2 questions

- **Pêle-mêle** : 2 questions

Tout comme pour le questionnaire des exercices de mathématiques, aucune cotation n'est donnée aux étudiants.

**L'ensemble du questionnaire se trouve à l'annexe C.**

#### Dernier questionnaire

Nous soumettons aux étudiants le même questionnaire d'auto-évaluation que le premier questionnaire afin qu'ils puissent éventuellement adapter leurs réponses après avoir passé les deux questionnaires "techniques".

#### **4.2.2 Création du questionnaire post-examen**

Nous réalisons ce questionnaire qui est soumis à l'ensemble des étudiants, qu'ils aient ou non répondu aux quatre questionnaires précédents.

Ce questionnaire demande d'abord aux étudiants s'ils ont participé aux questionnaires d'auto-évaluation et aux questionnaires "techniques" :

- Dans la négative : nous demandons à l'étudiant s'il a réussi son examen ou pas, la raison pour laquelle il n'a pas participé aux questionnaires précédents et s'il pense qu'il aurait pu identifier des lacunes en complétant les formulaires.
- Dans l'affirmative : nous demandons à l'étudiant s'il a réussi son examen ou pas, si les questionnaires lui ont permis d'identifier des lacunes ou de la matière à retravailler et ce qu'il souhaiterait ou conseilleraient voir dans un module d'aide à la réussite.

**L'ensemble du questionnaire se trouve à l'annexe D.**

### **4.3 Passation des questionnaires**

#### **4.3.1 Les premiers questionnaires**

Début décembre, les quatre premiers questionnaires sont soumis aux étudiants par un assistant du cours d'introduction à la programmation via la plateforme de l'université WebCampus.

La participation aux questionnaires est facultative mais vivement conseillée. Un rappel est fait par mail à l'ensemble des étudiants inscrits au cours via le secrétariat aux alentours du 20 décembre.

Lorsque l'étudiant a terminé le questionnaire de mathématiques, il peut, s'il le souhaite, repasser le questionnaire et vérifier ses réponses. En cas d'erreur, il a accès à la bonne réponse comme le montre la figure 8.

✖ Calcule  $D = 3A(B + 2C)$  \* 0 / 0

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & 8 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ -1 & 9 \end{bmatrix} \quad C = \begin{bmatrix} -1 & 0 \\ 0 & -2 \\ 1 & 1 \end{bmatrix}$$

$D = \begin{bmatrix} 6 & 18 \\ 12 & 243 \end{bmatrix}$ 

Option 1

$D = \begin{bmatrix} 90 & 73 \\ 261 & 219 \end{bmatrix}$ 

Option 2

$D = \begin{bmatrix} 6 & 87 \\ 12 & 168 \end{bmatrix}$ 

Option 3 ✖

$D = \begin{bmatrix} 6 & 87 \\ 12 & 313 \end{bmatrix}$ 

Option 4

Bonne réponse

Option 1

FIGURE 8 – Exemple de réponse erronée au questionnaire de mathématiques

Le questionnaire Python comprend des questions à choix multiples et des questions ouvertes. Au niveau des questions à choix multiples, l'étudiant peut consulter les réponses correctes comme pour le questionnaire de mathématiques.

En ce qui concerne les questions ouvertes, l'étudiant peut revoir la question où nous notons une explication. Celle-ci est développée sur le cheminement et la compréhension de chaque ligne de code pour permettre à l'étudiant de comprendre la réponse qui était attendue (comme le montre la figure 9).

Quelle sera la valeur de la variable "a" à la fin de l'exécution de ce code ? \*

```
1 a = 1
2 b = -1
3 a = a * b
4 a = a + b
5
6 print(a)
```

-2

**Commentaire**

*La réponse est -2.  
Successivement, nous avons a = 1 (ligne 1) et b = -1 (ligne 2).  
Ligne 3 :  $a = 1 * (-1)$ ; la valeur de a à ce moment-là vaut -1.  
Ligne 4 :  $a = (-1) + (-1) = -2$ ; la valeur de a est -1 suite à l'opération précédente, et nous additionnons b qui vaut -1.*

FIGURE 9 – Exemple de réponse ouverte expliquée au questionnaire Python

### 4.3.2 Le questionnaire post-examen

Afin d'inviter les étudiants à compléter le dernier questionnaire, nous avons réalisé une slide (voir figure 10) avec un lien et un QR-code, les deux renvoyant vers le questionnaire, qui a été partagée par notre promoteur lors de deux cours qu'il a dispensés aux étudiants courant du mois de février.

Nous avons également demandé au secrétariat d'envoyer un mail de rappel à l'ensemble des étudiants inscrits au cours d'introduction à la programmation vers la mi-mars avec une copie de la slide en pièce jointe.

## Mémoire - Réussir en informatique : Connaître ses faiblesses pour mieux y répondre

- Quatre questionnaires proposés mi-décembre dans le cadre du cours d'introduction à la programmation ont été complétés par certains d'entre vous (et nous vous en remercions).
- Un nouveau (et dernier) questionnaire à compléter (max. 5 min) même si les précédents ne l'ont pas été.
- Lien : <https://forms.gle/UnzWNQNnDHRJiXMt7>

- QR code :



Questions/remarques ?

[jeremy.dekinder@student.unamur.be](mailto:jeremy.dekinder@student.unamur.be)  
[paul.peytier@student.unamur.be](mailto:paul.peytier@student.unamur.be)



*Année académique 2022-2023*

FIGURE 10 – Slide réalisée pour le questionnaire post-examen

## 4.4 Collecte des résultats

### 4.4.1 Les premiers questionnaires

Pour les quatre premiers questionnaires, nous avons fixé une date limite au 02 janvier 2023 pour que les étudiants les complètent.

Après notre session d'examens de janvier, nous avons collecté les résultats.

Pour le questionnaire de mathématiques, étant donné qu'il s'agit d'un QCM, nous avons, pour chaque étudiant, reporté ses réponses pour chaque question dans un fichier Excel en y indiquant la réponse correcte, la réponse de l'étudiant ainsi que le degré de certitude de l'étudiant. A partir de là, en se basant sur l'échelle de score des degrés de certitudes (voir figure 3), nous avons attribué les notes correspondantes. Ceci nous permettra d'effectuer différents traitements sur les données collectées ; ce qui sera examiné à la section suivante.

Pour le questionnaire de programmation en langage Python, vu que celui-ci comporte des

questions de type QCM et des questions de type ouverte, nous avons collecté les résultats un peu différemment. En effet, pour les questions de type QCM, nous avons appliqué la même méthode que pour le questionnaire de mathématiques. Pour les questions ouvertes, nous avons examiné chaque réponse de l'étudiant afin de déterminer si celle-ci était correcte ou non.

Pour les questions demandant de la production de code, nous n'avons pas, en concertation avec notre copromotrice, tenu compte de l'indentation présente ou non au niveau du code. Chaque code fourni a également été vérifié dans l'IDE Pycharm afin d'en vérifier la compilation et l'exécution.

Ensuite, nous avons pu regrouper les résultats des questions dans un fichier Excel et attribuer les notes correspondantes.

En ce qui concerne les questionnaires d'auto-évaluation, nous avons encodé les données dans un fichier Excel afin de pouvoir les traiter.

#### **4.4.2 Le questionnaire post-examen**

Quinze jours après avoir effectué le rappel aux étudiants par mail via le secrétariat, nous avons remarqué que nous ne recevions plus de nouvelles contributions. Nous avons donc décidé de terminer la collecte des résultats à partir de ce moment-là.

Nous avons reporté l'ensemble des réponses dans un fichier Excel. Ensuite, nous avons scindé les réponses suivant que l'étudiant a complété ou non les premiers questionnaires.



## 5 Résultats

Dans cette section, nous présentons les données collectées lors des différents questionnaires ainsi que les traitements effectués sur celles-ci. L'interprétation des différents résultats obtenus sera, quant à elle, réalisée dans la section 6.

### 5.1 Questionnaires d'auto-évaluation

Le tableau 21 à l'annexe F reprend les données récoltées pour le premier questionnaire d'auto-évaluation. 29 étudiants ont répondu au premier questionnaire. Le second questionnaire n'ayant été complété que par 3 étudiants, nous ne reportons pas les données de celui-ci.

#### 5.1.1 Connaissances en algèbre

Dans le questionnaire, nous demandons aux étudiants d'évaluer leur niveau de connaissances en algèbre sur une échelle allant de 0 à 10; 0 représentant une connaissance nulle et 10 une parfaite connaissance de la matière.

Le tableau 3 reprend la répartition des étudiants suivant leur évaluation de leur niveau d'algèbre. Pour une meilleure visualisation, la figure 11 représente un histogramme reprenant ces valeurs.

Niveaux d'estimation de connaissance	Nombre d'étudiants	Pourcentages
0	1	3.45%
1	0	0.00%
2	3	10.34%
3	0	0.00%
4	3	10.34%
5	2	6.90%
6	2	6.90%
7	8	27.59%
8	4	13.79%
9	3	10.34%
10	3	10.34%
<b>TOTAL</b>	<b>29</b>	<b>100%</b>

TABLE 3 – Répartition des étudiants selon l'estimation de leur niveau d'algèbre

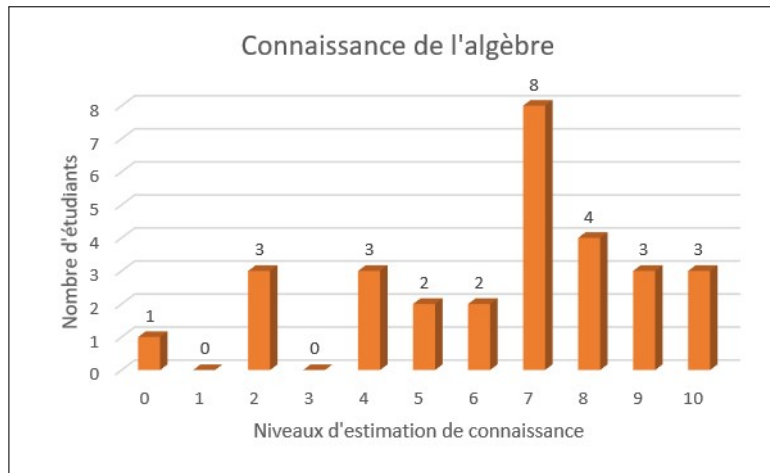


FIGURE 11 – Histogramme de la connaissance de l'algèbre selon l'auto-évaluation des étudiants

### 5.1.2 Connaissances en mathématiques discrètes

Dans le questionnaire, nous demandons aux étudiants d'évaluer leur niveau de connaissances en mathématiques discrètes sur une échelle allant de 0 à 10 ; 0 représentant une connaissance nulle et 10 une parfaite connaissance de la matière.

Le tableau 4 reprend la répartition des étudiants suivant leur évaluation de leur niveau de mathématiques discrètes. Pour une meilleure visualisation, la figure 12 représente un histogramme reprenant ces valeurs.

Niveaux d'estimation de connaissance	Nombre d'étudiants	Pourcentages
0	0	0.00%
1	1	3.45%
2	4	13.79%
3	3	10.34%
4	2	6.90%
5	2	6.90%
6	1	3.45%
7	7	24.14%
8	5	17.24%
9	3	10.34%
10	1	3.45%
<b>TOTAL</b>	<b>29</b>	<b>100%</b>

TABLE 4 – Répartition des étudiants selon l'estimation de leur niveau de mathématiques discrètes

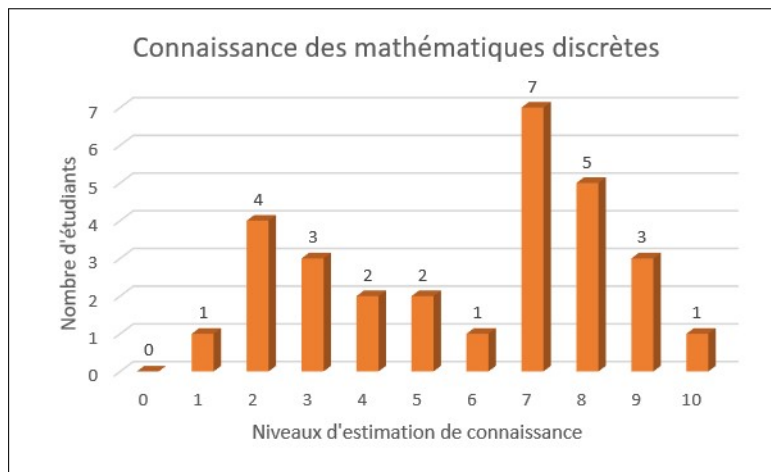


FIGURE 12 – Histogramme de la connaissance de l’algèbre selon l’auto-évaluation des étudiants

### 5.1.3 Niveau d’anglais

Le tableau 5 reprend la répartition des étudiants suivant leur évaluation de leur niveau d’anglais selon la norme CECRL. Pour une meilleure visualisation, la figure 13 représente un histogramme reprenant ces valeurs.

Niveau	Nombre d’étudiants	Pourcentages
<b>A0</b>	0	0.00%
<b>A1</b>	1	3.45%
<b>A2</b>	1	3.45%
<b>B1</b>	13	44.83%
<b>B2</b>	13	44.83%
<b>C1</b>	1	3.45%
<b>C2</b>	0	0.00%
<b>TOTAL</b>	<b>29</b>	<b>100%</b>

TABLE 5 – Répartition des étudiants selon leur niveau d’anglais

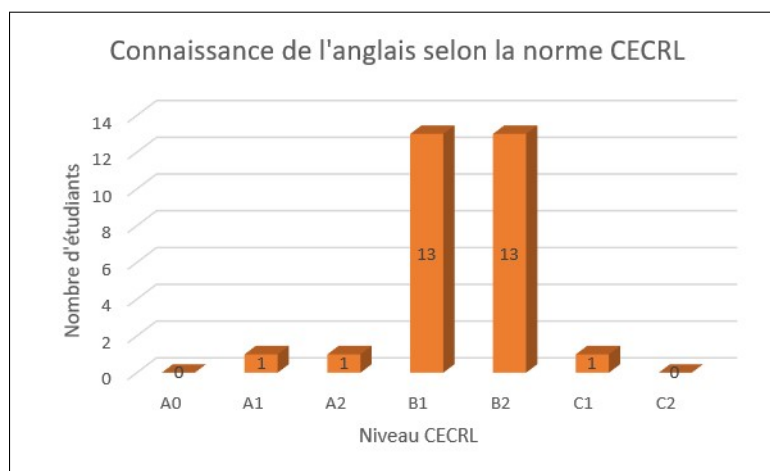


FIGURE 13 – Histogramme de la connaissance de l’anglais selon la norme CECRL

### 5.1.4 Connaissances en programmation Python

Dans le questionnaire, nous demandons aux étudiants d'évaluer leur niveau de connaissances en langage de programmation Python sur une échelle allant de 0 à 10; 0 représentant une connaissance nulle et 10 une parfaite connaissance de la matière.

Le tableau 6 reprend la répartition des étudiants suivant leur évaluation de leur niveau de langage de programmation Python. Pour une meilleure visualisation, la figure 14 représente un histogramme reprenant ces valeurs.

Niveaux d'estimation de connaissance	Nombre d'étudiants	Pourcentages
0	0	0.00%
1	0	0.00%
2	0	0.00%
3	2	6.90%
4	3	10.34%
5	3	10.34%
6	10	34.48%
7	6	20.69%
8	3	10.34%
9	1	3.45%
10	1	3.45%
<b>TOTAL</b>	<b>29</b>	<b>100%</b>

TABLE 6 – Répartition des étudiants selon l'estimation de leur niveau du langage de programmation Python

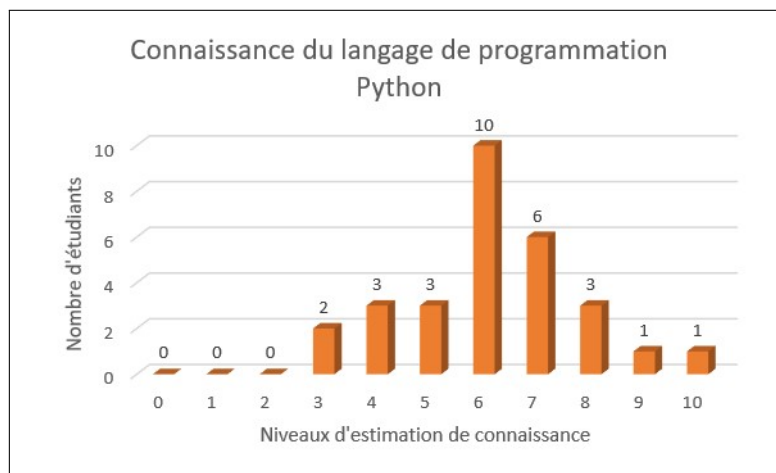


FIGURE 14 – Histogramme de la connaissance du langage de programmation Python selon l'auto-évaluation des étudiants

### 5.1.5 Connaissances d'autres langages de programmation

Enfin, nous demandons aux étudiants s'ils connaissaient un autre langage de programmation que Python. Le tableau 7 reprend les réponses aux questions et la figure 15 montre l'historgramme correspondant.

Autre langage de programmation connu	Nombre d'étudiants	Pourcentages
OUI	11	37.93%
NON	18	62.07%
<b>TOTAL</b>	<b>29</b>	<b>100%</b>

TABLE 7 – Connaissance d'autre(s) langage(s) de programmation

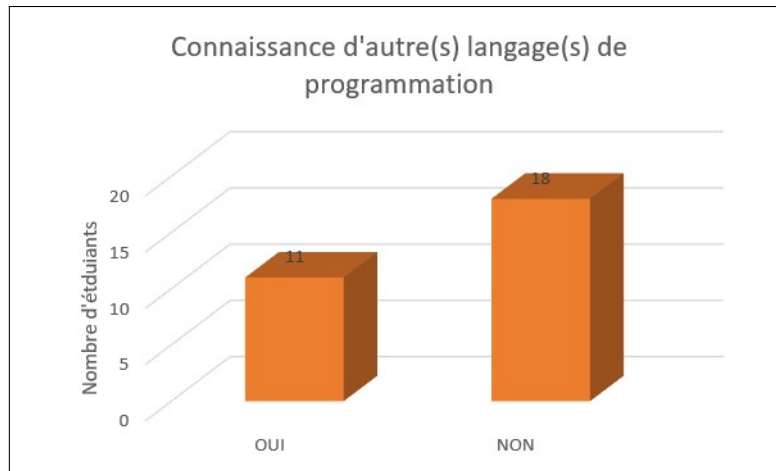


FIGURE 15 – Histogramme de la connaissance d'autre(s) langage(s) de programmation

## 5.2 Questionnaire de mathématiques

En appliquant la cotation suivant les degrés de certitude comme définie dans [31] et [32] (voir figure 3), nous avons noté les différents étudiants ayant rempli le questionnaire. Le tableau 8 reprend les résultats. L'ensemble des réponses des étudiants se trouvent à l'annexe G, dans les tableaux 22, 23, 24, 25, 26 et 27.

Cotations	ID-01	ID-02	ID-03	ID-04	ID-05	ID-06	ID-07	ID-8	ID-9	ID-10	ID-11	ID-12	ID-13	ID-14	ID-15	ID-16	ID-17	Moyennes
Q1	-20	20	20	20	2	16	20	20	20	19	20	18	0	20	20	0	20	13.824
Q2	20	20	19	20	16	3	20	20	20	19	20	-20	2	19	20	17	20	15.000
Q3	20	20	20	20	2	4	20	19	20	18	20	3	2	19	20	13	18	15.176
Q4	0	16	18	2	2	4	3	18	4	17	2	0	2	2	20	19	19	8.706
Q5	-20	20	19	19	0	17	20	20	20	19	17	20	17	0	4	19	20	13.588
Q6	-20	20	-20	0	18	-6	-20	18	-6	-20	-20	20	20	0	-20	18	20	0.118
Q7	-20	20	20	19	0	17	20	19	0	20	20	0	0	19	18	0	20	11.294
Q8-1	19	19	-6	16	2	2	-20	2	19	19	17	17	19	19	-6	3	-20	7.118
Q8-2	-6	-6	19	3	19	2	-20	2	19	-6	17	2	-6	-6	-6	3	20	2.941
Q9	20	20	19	18	18	18	20	18	18	19	17	19	20	20	20	18	20	18.941
Q10	20	20	20	20	3	3	20	4	20	19	20	2	20	18	19	18	20	15.647
Q11	-6	0	-20	18	17	3	19	13	17	0	2	3	19	2	17	18	-20	6.000
Q12	19	4	19	-20	-6	13	19	3	13	-6	-20	3	-20	4	0	3	-20	0.471
Q13	20	20	20	20	2	0	20	4	19	19	20	2	-20	0	20	2	20	11.059
Q14	13	-20	-6	-20	17	2	-20	4	19	-20	3	2	16	0	3	13	-20	-0.824
Q15	13	20	20	20	0	3	20	17	19	19	20	18	2	0	19	4	20	13.765
Q16	13	4	19	4	18	2	-6	4	18	2	4	2	2	13	13	4	-20	5.647
Q17	19	20	20	20	-20	2	20	18	18	19	20	20	19	18	19	0	20	14.824
Q18	4	20	19	20	20	18	20	4	20	20	18	20	17	19	4	4	20	15.706
Q19	4	20	20	4	20	0	20	4	19	19	20	20	3	19	4	13	20	13.471
Q20	4	20	20	20	20	18	20	4	19	20	20	20	16	19	4	4	20	15.765
Q21	4	13	20	17	20	16	20	4	18	4	20	-20	3	13	4	13	20	11.118
Q22	4	20	19	19	16	13	-6	4	13	13	17	17	16	18	4	4	-20	10.059
Q23	0	20	19	4	2	3	20	13	20	19	20	20	4	4	18	13	4	11.941
Q24	4	13	19	13	3	19	20	4	19	18	20	2	3	13	3	13	4	11.176
Q25	19	4	19	13	16	16	19	13	20	0	-20	20	16	4	2	4	4	9.941
Q26	16	20	19	20	2	4	20	4	20	19	20	20	16	4	4	4	4	12.706
TOTAL/540	163	387	394	329	229	212	308	277	445	308	334	250	208	280	247	244	233	285.176
TOTAL/20	6.037	14.333	14.593	12.185	8.481	7.852	11.407	10.259	16.481	11.407	12.370	9.259	7.704	10.370	9.148	9.037	8.630	10.562

TABLE 8 – Notes obtenues au questionnaire de mathématiques

### 5.2.1 Identification de la matière posant problème

Nous mettons en avant les questions dont la moyenne arithmétique est inférieure au score de 10/20.

Il s'agit des questions Q4, Q6, Q8-1, Q8-2, Q11, Q12, Q14, Q16 et Q25.

Au niveau des matières, cela correspond à :

- Q4 : algèbre
- Q6 : mathématiques discrètes - logique des propositions
- Q8-1 : mathématiques discrètes - logique des propositions
- Q8-2 : mathématiques discrètes - logique des propositions
- Q11 : mathématiques discrètes - logique des prédicats
- Q12 : mathématiques discrètes - logique des prédicats
- Q14 : mathématiques discrètes - calcul booléen
- Q16 : mathématiques discrètes - calcul booléen
- Q25 : mathématiques discrètes - codage-décodage

Nous pouvons donc présenter par matière le ratio de questions réussies par rapport au nombre total de questions. Ces données sont reprises au tableau 9 et les pourcentages de questions réussies par matière sont représentés sous forme d'histogramme à la figure 16.

Matières	Nombre de questions réussies	Nombre total de questions	Pourcentages
Algèbre	4	5	80.00%
Logique des propositions	2	5	40.00%
Logique des prédicats	2	4	50.00%
Calcul booléen	2	4	50.00%
Systèmes de numération de représentation	5	5	100.00%
Codage-décodage	3	4	75.00%

TABLE 9 – Ratio questions réussies/nombre total de questions par matière

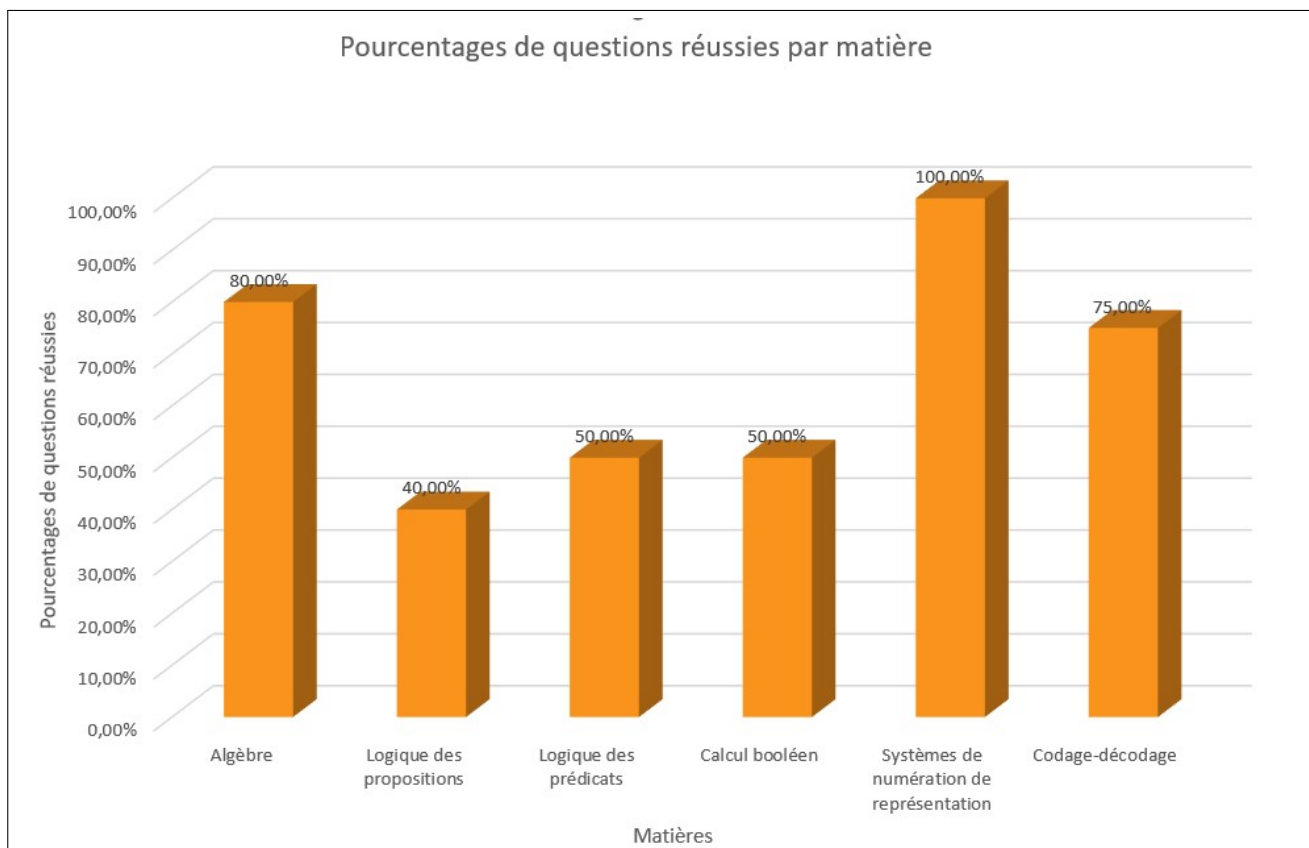


FIGURE 16 – Histogramme des pourcentages des questions réussies par matière

### 5.2.2 La centration

Afin déterminer la tendance de l'étudiant à se surestimer ou à se sous-estimer, nous calculons, sur base des résultats au questionnaire technique, l'indice de centration comme définit par J-L. GILLES dans [31] et [32] et expliqué à la section 2.6

Nous listons le nombre total de bonnes réponses par degré de certitude et le nombre total d'utilisations de chaque degré de certitude pour chaque étudiant. Le tableau 10 reprend ces données.

ID	DC0 - OK	DC0 - TOT	DC1 - OK	DC1 - TOT	DC2 - OK	DC2 - TOT	DC3 - OK	DC3 - TOT	DC4 - OK	DC4 - TOT	DC5 - OK	DC5 - TOT
ID-01	3	9	1	1	0	0	0	2	4	6	5	9
ID-02	2	5	1	1	0	0	0	1	1	2	17	18
ID-03	0	0	0	0	0	0	1	1	12	14	10	12
ID-04	2	5	1	2	1	2	2	3	3	3	10	12
ID-05	0	0	3	5	2	9	3	6	1	2	4	5
ID-06	2	5	3	8	2	7	3	5	1	2	0	0
ID-07	0	0	0	1	0	0	0	0	3	5	17	21
ID-08	3	14	0	1	1	3	4	4	2	2	3	3
ID-09	2	3	0	0	1	1	4	5	8	9	9	9
ID-10	1	2	0	0	1	2	2	4	12	14	3	5
ID-11	0	1	0	1	5	7	1	1	0	0	14	17
ID-12	0	0	0	3	2	8	2	4	1	1	9	11
ID-13	0	1	5	8	2	7	0	2	3	4	3	5
ID-14	3	7	0	0	0	2	3	8	7	8	2	2
ID-15	1	8	0	2	1	2	2	3	3	5	6	7
ID-16	6	13	0	3	1	2	4	7	2	2	0	0
ID-17	0	4	0	0	0	0	1	1	1	1	15	21

TABLE 10 – Nombre total de bonnes réponses ( $DC_{\text{degré de certitude}} - \text{OK}$ ) et le nombre total d'utilisation du degré de certitude ( $DC_{\text{degré de certitude}} - \text{TOT}$ )

Ensuite, nous calculons les erreurs de centration pour chaque étudiant.

Par exemple, pour l'étudiant ID-01, le calcul de sa certitude moyenne est :

$$CM = \frac{0.125*9+0.375*1+0.6*0+0.775*2+0.9*6+0.975*9}{27} * 100 = 64$$

Le taux d'exactitude se calcule simplement comme le nombre de réponses correctes divisé par le nombre total des questions ; le tout multiplié par 100.

De nouveau, pour l'étudiant ID-01 :

$$TE = \frac{3+1+0+0+4+5}{27} * 100 = 48$$

Enfin, l'erreur de centration est la différence entre la certitude moyenne et le taux d'exactitude.

Ce qui donne pour l'étudiant ID-01 :

$EC = 64 - 48 = 16$  ; ce qui indique, suivant l'échelle de Gilles (voir figure 4) que l'étudiant a une surestimation assez forte.

Le tableau 11 reprend le calcul des erreurs de centration de chaque étudiant et s'il se surestime ou se sous-estime.



ID	CM	TE	EC	Auto-estimation
ID-01	64	48	16	Surestimation assez forte
ID-02	78	78	0	Auto-estimation excellente
ID-03	93	85	8	Surestimation très légère
ID-04	71	70	1	Auto-estimation excellente
ID-05	69	48	21	Surestimation forte
ID-06	50	41	9	Surestimation très légère
ID-07	94	74	20	Surestimation forte
ID-08	44	48	-4	Auto-estimation excellente
ID-09	80	89	-9	Sous-estimation très légère
ID-10	82	70	12	Surestimation légère
ID-11	82	74	8	Surestimation très légère
ID-12	76	52	24	Surestimation forte
ID-13	64	48	16	Surestimation assez forte
ID-14	65	56	9	Surestimation très légère
ID-15	61	48	13	Surestimation légère
ID-16	41	48	-7	Sous-estimation très légère
ID-17	84	63	21	Surestimation forte

TABLE 11 – Erreurs de centrage pour le questionnaire de mathématiques et auto-estimations correspondantes

### 5.3 Questionnaire de programmation en langage Python

Tout comme pour le questionnaire de mathématiques, nous avons appliqué la cotation suivant les degrés de certitude définie par Gilles ([31] et [32]). Le tableau 12 reprend les résultats des étudiants. L'ensemble des réponses des étudiants se trouvent à l'annexe H, dans les tableaux 28, 29, 30 et 31.

Cotations	ID-01	ID-02	ID-03	ID-04	ID-05	ID-06	ID-07	ID-08	ID-09	ID-10	ID-11	ID-12	ID-13	ID-14	ID-18	ID-19	ID-20	ID-21	ID-22	ID-23	ID-24	ID-25	ID-26	ID-27	ID-28	ID-29	Moyennes	
Q1	20	20	20	20	-6	20	19	19	20	19	20	-6	19	19	20	20	20	20	4	2	20	20	-6	20	20	20	15.500	
Q2	20	20	20	20	19	20	19	19	19	19	20	20	20	19	20	20	20	20	20	20	20	20	20	18	20	20	19.692	
Q3	20	-20	20	20	18	20	-20	20	20	19	20	20	-6	19	20	20	20	20	20	20	20	-20	20	20	20	-20	12.692	
Q4	20	0	20	20	-20	20	20	18	20	19	0	-20	20	19	20	20	20	19	20	0	20	2	-20	-20	20	-20	9.115	
Q5	20	2	20	20	18	20	20	19	20	20	20	20	20	20	19	20	20	20	20	20	20	20	20	20	20	20	19.154	
Q6	20	20	19	20	-6	20	20	19	19	0	20	2	19	2	20	20	19	19	3	-20	19	20	20	19	20	20	14.346	
Q7	-20	20	17	-6	19	-6	19	17	2	-6	20	20	-6	2	20	20	20	18	16	20	20	-20	20	19	20	20	10.962	
Q8	20	-20	20	19	-6	20	20	17	20	18	18	0	19	2	20	20	20	2	4	0	20	18	20	20	19	-20	11.923	
Q9-1	20	20	20	19	20	20	20	18	18	19	17	17	20	19	20	20	20	19	19	0	19	19	0	19	19	20	20	18.538
Q9-2	20	20	20	19	-20	20	20	18	18	19	2	2	20	19	20	20	19	-6	0	19	-6	-20	19	20	20	20	12.388	
Q10	-20	18	20	20	4	-20	20	3	20	-6	20	2	-6	0	-20	-20	20	0	4	4	-20	4	20	20	20	0	4.115	
Q11	4	-20	20	18	4	20	-20	4	20	19	2	2	-6	0	20	20	20	4	4	4	-20	4	-20	-6	-6	4	3.654	
Q12	20	20	20	20	20	20	20	19	19	20	20	13	20	18	20	20	20	18	20	2	18	19	20	19	20	-20	17.115	
Q13	20	20	20	18	4	20	20	17	20	19	20	3	19	18	20	20	20	19	-6	19	0	-6	20	20	20	20	15.538	
Q14	-20	17	19	0	0	19	20	0	0	19	-6	3	-6	18	-20	20	-20	19	-20	0	0	2	20	20	19	19	5.462	
Q15	20	20	-6	20	17	20	20	3	19	18	16	3	19	19	19	20	19	19	13	17	18	2	20	-6	19	20	14.923	
Q16	20	19	19	-6	4	-20	-20	4	19	19	-20	2	0	0	20	20	20	2	4	4	-20	2	20	-20	-20	0	2.709	
Q17-1	19	18	18	17	4	20	17	2	18	17	0	3	17	18	20	20	20	0	4	4	4	3	-6	20	20	4	11.577	
Q17-2	19	18	18	17	4	20	17	2	18	2	0	3	17	0	-20	-20	20	0	4	4	4	3	-6	20	20	4	7.231	
Q18	4	20	-6	17	4	18	17	17	2	0	20	2	18	18	-20	-20	20	-6	4	17	18	2	20	4	-20	19	7.269	
Q19	4	-20	0	4	4	-20	19	4	17	-6	-6	2	-6	17	4	20	20	18	4	4	-6	4	-20	4	4	-6	2.423	
Q20	0	4	20	20	0	18	0	2	2	20	4	3	17	19	20	20	20	2	-20	-6	-6	19	0	19	19	0	8.308	
Q21	4	20	19	19	4	19	17	4	20	19	20	3	3	0	20	20	20	-6	4	4	-6	4	20	19	20	4	11.308	
TOTAL /420	254	236	377	355	109	308	304	265	370	306	247	119	251	284	283	340	418	265	139	139	181	135	202	287	334	148	256.000	
TOTAL /20	12.095	11.238	17.952	16.905	5.190	14.667	14.476	12.619	17.619	14.571	11.762	5.667	11.952	13.524	13.476	16.190	19.905	12.619	6.619	6.619	8.619	6.429	9.619	13.667	15.905	7.048	12.190	

TABLE 12 – Notes obtenues au questionnaire de programmation en langage Python

### 5.3.1 Identification de la matière posant problème

Nous mettons en avant les questions dont la moyenne arithmétique est inférieure au score de 10/20.

Il s'agit des questions Q4, Q10, Q11, Q14, Q16, Q17-2, Q18, Q19 et Q20.

Au niveau des matières, cela correspond à :

- Q4 : les conditions
- Q10 : les boucles
- Q11 : les boucles
- Q14 : les structures de données
- Q16 : les structures de données
- Q17-2 : les structures de données
- Q18 : les fichiers
- Q19 : les fichiers
- Q20 : questions pèle-mêle

Nous pouvons donc présenter par matière le ratio de questions réussies par rapport au nombre total de questions. Ces données sont reprises au tableau 13 et les pourcentages de questions réussies par matière sont représentés sous forme d'histogramme à la figure 17.

Matières	Nombre de questions réussies	Nombre total de questions	Pourcentages
<b>Variables</b>	2	2	100.00%
<b>Conditions</b>	2	3	66.67%
<b>Fonctions</b>	3	3	100.00%
<b>Boucles</b>	1	3	33.33%
<b>Structures de données</b>	4	7	57.14%
<b>Fichiers</b>	0	2	0.00%
<b>Pèle-mêle</b>	1	2	50.00%

TABLE 13 – Ratio questions réussies/nombre total de questions par matière

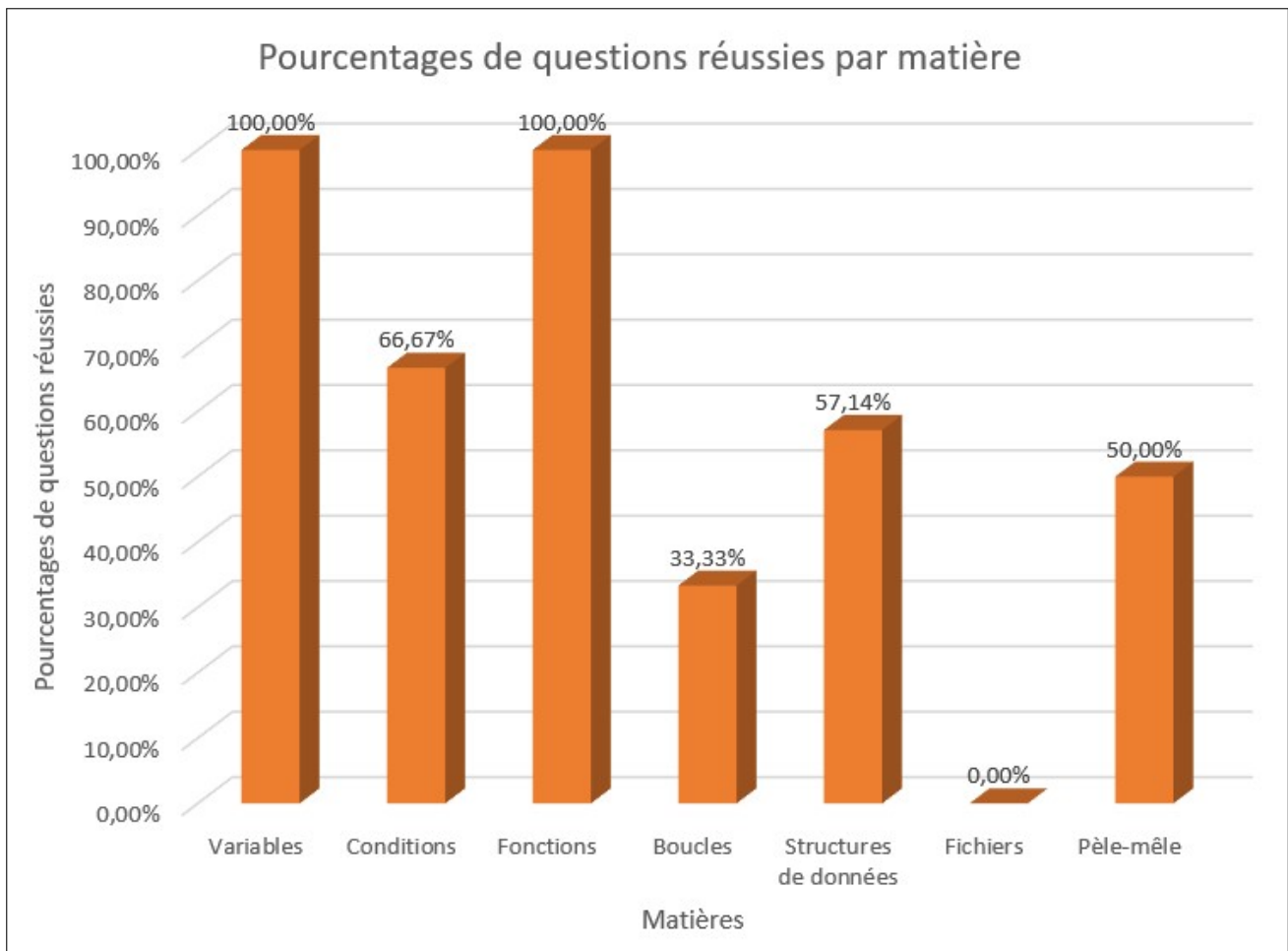


FIGURE 17 – Histogramme des pourcentages des questions réussies par matière

### 5.3.2 La centration

Tout comme pour le calcul de l'indice de centration pour le questionnaire de mathématiques, nous reprenons la formule

$$EC = CM - TE$$

définie par Gilles dans [31] et [32].

Nous reprenons également les valeurs centrales des intervalles des degrés de certitude repris au tableau 1

Nous listons ensuite le nombre total de bonnes réponses par degré de certitude et le nombre total d'utilisations de chaque degré de certitude pour chaque étudiant. Le tableau 14 reprend ces données.

ID	DC0 - OK	DC0 - TOT	DC1 - OK	DC1 - TOT	DC2 - OK	DC2 - TOT	DC3 - OK	DC3 - TOT	DC4 - OK	DC4 - TOT	DC5 - OK	DC5 - TOT
ID-01	0	4	0	0	0	0	0	1	2	2	13	16
ID-02	0	1	0	0	1	2	3	4	1	1	11	15
ID-03	0	0	0	0	1	1	2	3	4	6	13	13
ID-04	0	1	0	0	3	3	2	3	4	6	10	10
ID-05	0	9	0	0	1	1	2	4	2	5	2	4
ID-06	0	0	0	0	0	0	2	2	2	3	15	18
ID-07	0	0	0	0	4	4	0	1	4	4	11	14
ID-08	0	4	0	2	4	7	3	4	5	5	1	1
ID-09	0	0	0	0	1	4	4	5	5	5	9	9
ID-10	0	0	0	0	1	2	2	4	11	14	3	3
ID-11	0	1	1	1	1	3	1	4	0	2	11	12
ID-12	1	1	0	7	1	8	0	1	0	1	4	5
ID-13	0	0	0	1	3	3	1	2	5	11	6	6
ID-14	0	0	0	0	1	4	5	10	9	9	0	0
ID-18	0	1	0	0	0	0	0	0	1	1	17	21
ID-19	0	0	0	0	0	0	0	0	0	0	20	23
ID-20	0	0	0	0	0	0	0	0	2	2	20	21
ID-21	0	1	0	0	0	3	3	6	7	9	4	4
ID-22	1	11	1	2	0	0	0	0	1	3	5	7
ID-23	0	7	0	0	2	4	0	5	1	2	4	5
ID-24	0	2	0	0	0	0	3	5	3	6	7	10
ID-25	0	4	0	2	0	5	1	1	3	5	4	6
ID-26	0	0	0	0	0	0	0	1	0	3	15	19
ID-27	0	2	0	0	0	0	1	1	7	9	9	11
ID-28	0	1	0	0	0	0	0	0	4	5	15	17
ID-29	0	4	0	0	0	0	0	3	2	3	9	13

TABLE 14 – Nombre total de bonnes réponses (DCdegré de certitude - OK) et le nombre total d'utilisation du degré de certitude (DCdegré de certitude - TOT)

En appliquant la formule l'indice de centration, nous obtenons le tableau 15 qui reprend le calcul des erreurs de centration de chaque étudiant et s'il se surestime ou se sous-estime.

<b>ID</b>	<b>CM</b>	<b>TE</b>	<b>EC</b>	<b>Auto-estimation</b>
<b>ID-01</b>	81	65	16	Surestimation assez forte
<b>ID-02</b>	87	70	17	Surestimation assez forte
<b>ID-03</b>	91	87	4	Auto-estimation excellente
<b>ID-04</b>	84	83	1	Auto-estimation excellente
<b>ID-05</b>	58	30	28	Surestimation très forte
<b>ID-06</b>	95	83	12	Surestimation légère
<b>ID-07</b>	89	83	6	Surestimation très légère
<b>ID-08</b>	61	57	4	Auto-estimation excellente
<b>ID-09</b>	85	83	2	Auto-estimation excellente
<b>ID-10</b>	86	74	12	Surestimation légère
<b>ID-11</b>	82	61	21	Surestimation forte
<b>ID-12</b>	61	26	35	Surestimation extrême
<b>ID-13</b>	85	65	20	Surestimation forte
<b>ID-14</b>	79	65	14	Surestimation légère
<b>ID-18</b>	93	78	15	Surestimation assez forte
<b>ID-19</b>	98	87	11	Surestimation légère
<b>ID-20</b>	97	96	1	Auto-estimation excellente
<b>ID-21</b>	81	61	20	Surestimation forte
<b>ID-22</b>	51	35	16	Surestimation assez forte
<b>ID-23</b>	60	30	30	Surestimation extrême
<b>ID-24</b>	84	57	27	Surestimation très forte
<b>ID-25</b>	67	35	32	Surestimation extrême
<b>ID-26</b>	96	65	31	Surestimation extrême
<b>ID-27</b>	86	74	12	Surestimation légère
<b>ID-28</b>	92	83	9	Surestimation très légère
<b>ID-29</b>	79	48	31	Surestimation extrême

TABLE 15 – Erreurs de centration pour le questionnaire de programmation en langage Python et auto-estimations correspondantes

## 5.4 Questionnaire post-examen

L'ensemble des réponses au questionnaire se trouve aux tableaux 32 et 33 à l'annexe I

Sur les 20 étudiants qui ont répondu au questionnaire post-examen, 9 ont passé les questionnaires précédents et 11 non.

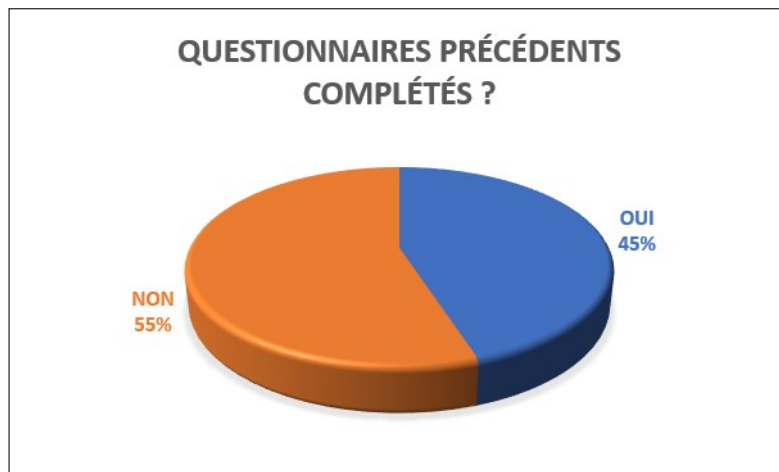


FIGURE 18 – Graphe en secteurs montrant la répartition des étudiants ayant passé ou non les questionnaires précédents

#### 5.4.1 Etudiants ayant complété les questionnaires précédents

Les 9 étudiants ont réussi leur examen d'introduction à la programmation et pour 8 d'entre-eux, les questionnaires ont permis d'identifier des lacunes ou de la matière à revoir.

Au niveau du développement d'un module pour aider à la réussite, voici les conseils/souhaits que les étudiants ont émis :

- Tester le code donné en réponse
- Questions-type d'examens
- Plus d'exercices
- Explications simplifiées sur les matières pour lesquelles l'étudiant a des lacunes
- Partie sur les fichiers
- Partie sur la récursivité

#### 5.4.2 Etudiants n'ayant pas complété les questionnaires précédents

Sur les 11 étudiants, 9 ont réussi leur examen d'introduction à la programmation. 6 d'entre-eux pensent que le fait de réaliser les questionnaires leur aurait permis d'identifier des lacunes.

Enfin, nous avons également demandé la raison pour laquelle ils n'ont pas réalisé les questionnaires précédents. Pour la majorité, ils n'ont pas vu qu'il y avait des questionnaires. Le graphe repris à la figure 19 montre les différentes raisons pourquoi les questionnaires n'ont pas été complétés.

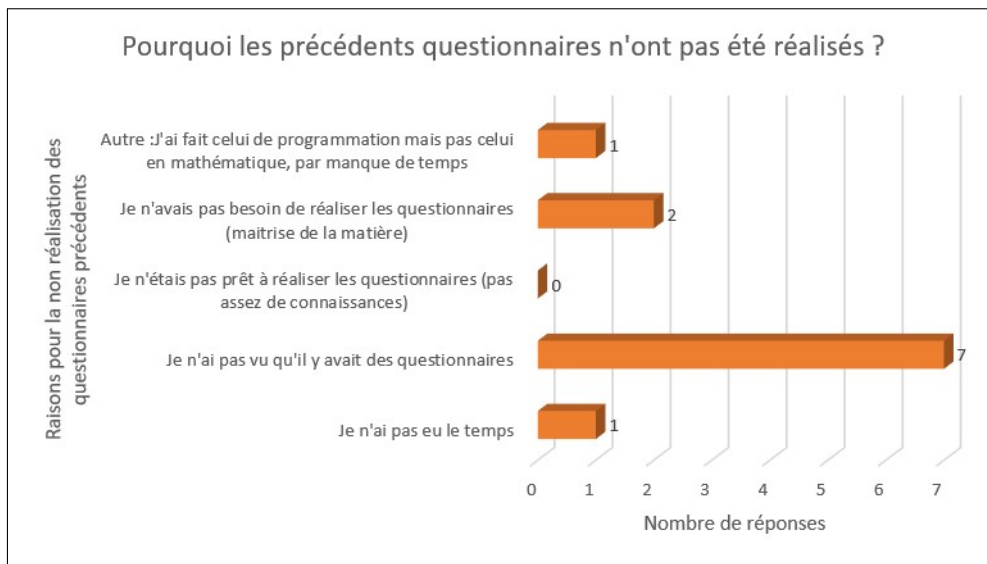


FIGURE 19 – Graphe montrant les raisons pour lesquelles les précédents questionnaires n’ont pas été complétés

## 6 Discussion

Dans cette section, en repartant de nos questions de recherche, nous discutons des résultats collectés et regardons si les hypothèses que nous avons émises sont vérifiées ou non.

### 6.1 QR1 : Quelles faiblesses dans l'apprentissage de la programmation l'auto-évaluation permet-elle d'identifier ?

Pour répondre à cette question de recherche, nous allons discuter des résultats collectés suite à la passation des questionnaires.

Nous examinons d'abord les points de matière identifiés comme problématiques. Ensuite, nous nous penchons sur la centration et l'auto-évaluation.

Enfin, nous répondons à notre première question de recherche.

#### 6.1.1 Matières problématiques

Les résultats aux différents questionnaires ont permis de mettre en évidence différents aspects concernant la compréhension et la maîtrise des mathématiques et de la programmation chez les étudiants.

En reprenant les matières où nous avons identifié des lacunes, nous examinons chaque question en détail.

##### Les mathématiques

Au niveau des mathématiques, en se basant sur les questions ratées énoncées à la section 5.2.1 et sur le tableau 9 nous examinons en détail les questions pour lesquelles la matière réussie est inférieure ou égale à 50% :

N° de question	Matières	Bonnes réponses	Mauvaises réponses	Pourcentage de réussite
Q6	Logique des propositions	7	10	41.18%
Q8-2	Logique des propositions	4	13	23.53%
Q11	Logique des prédicats	8	9	47.06%
Q12	Logique des prédicats	5	12	29.41%
Q14	Caclul booléen	5	12	29.41%
Q16	Caclul booléen	6	11	35.29%

TABLE 16 – Taux de réussite aux questions problématiques sans degrés de certitude

Nous constatons qu'indépendamment de la prise en compte des degrés de certitude, les matières reprises ci-dessus posent problème aux étudiants.

Nous décidons ensuite de nous baser sur les catégories du vecteur des degrés de connaissances (table 2) pour les réponses des étudiants aux questions reprises dans la table 16. Nous obtenons



la table 17 :

N° de question	Réponses correctes			Réponses incorrectes		
	DC0-DC1	DC2-DC3	DC4-DC5	DC0-DC1	DC2-DC3	DC4-DC5
	Compétence ignorée	Compétence partielle	Compétence assurée	Ignorance reconnue	Ignorance partielle	Ignorance ignorée
Q6	0.00%	17.65%	23.53%	0.00%	11.76%	<b>47.06%</b>
Q8-2	0.00%	11.76%	17.65%	11.76%	17.65%	<b>41.18%</b>
Q11	5.88%	<b>29.41%</b>	11.76%	11.76%	23.53%	17.65%
Q12	11.76%	0.00%	17.65%	29.41%	5.88%	<b>35.29%</b>
Q14	17.65%	5.88%	5.88%	17.65%	17.65%	<b>35.29%</b>
Q16	17.65%	11.76%	5.88%	<b>29.41%</b>	23.53%	11.76%
Moyenne	8.82%	12.75%	13.73%	16.67%	16.67%	<b>31.37%</b>

TABLE 17 – Représentation des pourcentages des réponses par rapport aux catégories du vecteur des degrés de connaissance pour les questions problématiques en mathématiques

En examinant la table 17, nous remarquons directement que pour la majorité des questions (Q6, Q8-2, Q12 et Q14), une majorité d'étudiants a une ignorance ignorée, ce qui signifie que ceux-ci ont fourni une réponse erronée mais en étant très sûr d'eux.

Au niveau de la Q16, dans ce cas, la majorité des étudiants ont une ignorance reconnue, ce qui traduit par le fait qu'ils étaient conscients de leur manque de connaissance ou de compréhension du sujet.

Enfin, au niveau de la Q11, c'est la catégorie de compétence partielle qui ressort. Les étudiants ont une certaine connaissance ou compréhension du sujet mais ils ne sont pas certains de leur réponse.

Nous pouvons donc émettre l'hypothèse qu'en ce qui concerne les questions ratées, les étudiants ne les ont pas réussies soit parce qu'ils ont trop confiance en eux par rapport à la matière, soit parce qu'ils n'ont pas assimilé celle-ci.

### Le langage de programmation Python

En ce qui concerne le langage de programmation Python, nous nous basons également sur les questions ratées énoncées à la section 5.3.1 et sur le tableau 13 et examinons en détail les questions pour lesquelles la matière réussie est inférieure ou égale à 50% :

N° de question	Matières	Bonnes réponses	Mauvaises réponses	Pourcentage de réussite
<b>Q10</b>	Boucles	10	16	38.46%
<b>Q11</b>	Boucles	8	18	30.77%
<b>Q20</b>	Boucles (pèle-mêle)	12	14	46.15%
<b>Q18</b>	Fichiers	13	13	50.00%
<b>Q19</b>	Fichiers	6	20	23.08%

TABLE 18 – Taux de réussite aux questions problématiques sans degrés de certitude

Tout comme pour les mathématiques, nous remarquons que les matières reprises ci-dessus

posent problème et ce, sans utiliser les degrés de certitude.

En ventilant les catégories du vecteur des degrés de connaissance, nous obtenons la table 19 :

N° de question	Réponses correctes			Réponses incorrectes		
	DC0-DC1	DC2-DC3	DC4-DC5	DC0-DC1	DC2-DC3	DC4-DC5
	Compétence ignorée	Compétence partielle	Compétence assurée	Ignorance reconnue	Ignorance partielle	Ignorance ignorée
Q10	0.00%	3.85%	<b>34.62%</b>	19.23%	15.38%	26.92%
Q11	0.00%	3.85%	26.92%	<b>30.77%</b>	11.54%	26.92%
Q20	0.00%	11.54%	<b>34.62%</b>	11.54%	30.77%	11.54%
Q18	0.00%	<b>30.77%</b>	19.23%	15.38%	15.38%	19.23%
Q19	0.00%	11.54%	11.54%	<b>38.46%</b>	7.69%	30.77%
Moyenne	0.00%	12.31%	<b>25.38%</b>	23.08%	16.15%	23.08%

TABLE 19 – Représentation des pourcentages des réponses par rapport aux catégories du vecteur des degrés de connaissance pour les questions problématiques en programmation

A l'examen de la table 19, nous remarquons que les résultats sont plus dispersés que pour les mathématiques.

Néanmoins, nous relevons qu'en moyenne, environ un quart des réponses présentent une compétence assurée, c'est-à-dire que les étudiants ont répondu correctement en étant sûrs de leurs réponses, qu'environ un quart des réponses présentent une ignorance reconnue et qu'un quart des réponses présentent une ignorance ignorée. Le dernier quart se répartissant entre la compétence partielle et l'ignorance partielle.

Nous pouvons donc également émettre la même hypothèse que pour les mathématiques en ce qui concerne les questions ratées ; à savoir que les étudiants ne les ont pas réussies, soit parce qu'ils ont trop confiance en eux par rapport à la matière, soit parce qu'ils n'ont pas assimilé la matière.

### Feedback du questionnaire Post-Examen

Pour le questionnaire Post-Examen, nous avons traité deux possibilités :

- Les étudiants qui ont passé les premiers questionnaires.
- Les étudiants qui n'ont pas présenté les premiers questionnaires.

Sur les 10 étudiants ayant répondu aux premiers questionnaires, 9 d'entre-eux signalent que les questionnaires d'auto-évaluation et techniques leurs ont permis d'identifier des lacunes.

En ce qui concerne les étudiants n'ayant pas participé aux premiers questionnaires, 6 étudiants sur 11 pensent qu'ils auraient pu identifier des lacunes en répondant aux questionnaires proposés.

Ceci nous conforte dans le fait que l'auto-évaluation permet aux étudiants d'identifier des lacunes dans les matières à assimiler.

### 6.1.2 Centration et auto-évaluation

En choisissant d'utiliser des questionnaires avec degrés de certitude, nous voulions examiner la manière dont les étudiants s'auto-évaluent et plus précisément s'ils se surestiment ou se sous-estiment.

En nous basant sur les tableaux 11 et 15 qui reprennent les erreurs de centration et les auto-estimations correspondantes pour les questionnaires techniques de mathématiques et du langage de programmation Python, nous pouvons identifier les étudiants qui ont trop confiance en leurs connaissances.

Nous remarquons également que les étudiants ont plutôt tendance à se surestimer.

### 6.1.3 Faiblesses identifiées

Dans notre première question de recherche, nous émettions l'hypothèse que l'utilisation du facteur d'auto-évaluation dans le cadre d'un cours d'introduction à la programmation permet d'identifier des faiblesses auprès des étudiants telles que des lacunes dans la connaissance de la matière mais aussi des problèmes dans l'appropriation de cette matière ou encore des difficultés à résoudre des problèmes.

Nous avons pu voir dans la sous-section 6.1.1 que ce facteur d'auto-évaluation nous a permis d'identifier des points de matière problématiques, que ce soit au niveau des mathématiques (logique des propositions, logique des prédicats et calcul booléen) ou de la programmation en langage Python (boucles et fichiers).

De plus, en programmation en langage Python, nous avons remarqué que la matière relative aux boucles était vraiment problématique. Ceci a déjà été relevé dans le cadre de l'étude effectuée dans [21] (comme il en a été fait mention dans notre revue de littérature).

Une deuxième hypothèse était que le facteur d'auto-évaluation permet également de détecter des problèmes dans la manière dont les étudiants estiment leur niveau de connaissance.

Dans la sous-section précédente, nous avons également vérifié cette hypothèse.

## 6.2 QR2 : Quelles améliorations peuvent être envisagées suite à l'intégration de l'auto-évaluation dans l'apprentissage de la programmation ?

Au niveau de cette seconde question de recherche, nous n'avons pas émis d'hypothèses. Nous allons plutôt explorer différentes pistes d'améliorations utilisant le facteur d'auto-évaluation.

### **6.2.1 Création d'un outil et/ou de modules d'apprentissage - Inginius**

Lors du choix du sujet de ce mémoire, nous avons émis le souhait d'éventuellement développer un outil permettant aux étudiants de s'entraîner sur certaines matières afin d'évaluer leurs compétences.

En discutant de ce point avec notre promoteur, il s'avère qu'un outil similaire a été développé par l'Université catholique de Louvain. Il s'agit de la plateforme Inginius qui contient différents modules d'apprentissage (et notamment des modules concernant l'apprentissage de la programmation).

Actuellement, l'Université de Namur est occupée à préparer, éventuellement, le déploiement de cet outil dans WebCampus. Nous avons donc pu prendre contact avec une des personnes s'occupant de la réalisation des tests et lui avons soumis nos questionnaires afin de vérifier si l'ajout de questionnaires similaires pouvaient être envisagé. Il nous a répondu par l'affirmative.

Aussi, en examinant les réponses du questionnaire post-examen, on remarque que les étudiants sont demandeurs de ce type d'outil afin de pouvoir s'exercer sur différents exercices mais aussi de pouvoir compiler leur code directement afin de vérifier si celui s'exécute correctement.

### **6.2.2 Aide à destination des professeurs ou des assistants**

Lors de notre revue de littérature, nous avons relevé dans [14], [15] et [16] que l'utilisation d'outils numériques permet d'identifier les étudiants à risque mais aussi des matières problématiques.

En développant des modules complémentaires d'exercices tout en incluant l'utilisation de l'auto-évaluation et en permettant au corps professoral d'avoir un retour sur les résultats des étudiants aux exercices proposés, le professeur ou ses assistants pourraient identifier si de la matière pose problème aux étudiants ou si ceux-ci estiment maîtriser la matière alors que ce n'est pas le cas.

Les enseignants pourraient alors adapter leurs cours en fonction des lacunes et/ou besoins identifiés.

D'un autre côté, en ayant un retour sur les résultats, les étudiants à risque pourraient être repérés de façon précoce et des mesures pourraient éventuellement être mises en place pour les amener vers la réussite.

### **6.2.3 Feedback vers les étudiants**

En s'entraînant sur des modules d'exercices, l'étudiant pourrait, de lui-même, se rendre compte des matières qui lui posent problème.

De plus, en intégrant les degrés de certitude et le calcul de l'indice de centration dans les modules, nous pourrions directement attirer l'attention de l'étudiant si ce dernier surestime ou sous-estime ses connaissances. Cela lui permettrait de corriger sa manière d'évaluer ses connaissances.

## 7 Biais

Dans cette section, nous souhaitons émettre quelques remarques par rapport aux résultats de nos recherches ; remarques qui *pourraient* avoir une influence sur ces résultats.

Dans un premier temps, nous tenons à signaler un faible taux de participation des étudiants à notre recherche.

- 29 étudiants ont complété le premier questionnaire d’auto-évaluation
- 17 étudiants ont complété le questionnaire technique de mathématiques
- 26 étudiants ont complété le questionnaire technique de programmation en langage Python
- 3 étudiants ont complété le second questionnaire d’auto-évaluation
- 20 étudiants ont complété le questionnaire post-examen

Certains auteurs ont déjà relevé des faibles taux de participation et ont essayé d’en expliquer les raisons.

Les auteurs de [30] expliquent que, selon eux, les étudiants consacrent moins de temps à des questionnaires d’auto-évaluation qu’à l’apprentissage de leurs cours et autres travaux académiques qui contribuent directement à leurs notes.

Dans [45], les auteurs signalent qu’ils ont demandé aux étudiants d’indiquer le numéro de carte d’étudiant et que cela a pu inciter certains d’entre-eux à ne pas participer à l’étude. Dans notre recherche, nous demandons également aux étudiants d’encoder leur numéro de carte d’étudiant afin de pouvoir faire le lien entre les différents questionnaires. Bien que nous ayons insisté sur l’anonymisation des réponses, le fait d’encoder le numéro de carte d’étudiant a pu décider certains à ne pas participer.

Ensuite, nos questionnaires n’étaient pas obligatoires. Comme le souligne les auteurs de [23], l’auto-évaluation est un processus d’analyse exigeant et difficile. Dès lors, il est important que l’étudiant s’implique volontairement dans cette démarche pour que celle-ci soit efficace. Sans cette volonté, il existe un risque que l’étudiant ne participe pas à cet exercice. Nos premiers questionnaires ayant été soumis au début de la période du blocus préparatoire à la session d’examens de janvier, les étudiants pourraient avoir préféré se concentrer sur l’étude de leurs examens plutôt que sur la réalisation des questionnaires.

Il ressort aussi du questionnaire post-examen que les étudiants n’ont pas complété les premiers questionnaires par manque de temps (ID-20 et ID-21) ou qu’ils n’avaient pas vu qu’il y avait des questionnaires (ID-31, ID-34, ID-35, ID-41, ID-44, ID-47 et ID-49).

Par ailleurs, bien que nous ayons déjà relevé des faiblesses, les résultats obtenus pourraient être plus positifs que la réalité. En effet, comme le signalent les auteurs de [30], les étudiants les plus performants seraient plus enclins à participer aux questionnaires.

Enfin, nous avons relevé des irrégularités dans les réponses reçues du questionnaire post-examen : ID-20 et ID-21 indiquent qu'ils n'ont pas présenté les questionnaires mais ils ont tous les deux présenté le questionnaire Python ; ID-33, ID-36, ID-39, ID-40 et ID-43 indiquent avoir passé les questionnaires mais ne sont pas dans repris dans les étudiants ayant validé les questionnaires (ont peut-être réalisé les questionnaires mais ne les ont pas validés/envoyés) ; idem pour ID-42 qui a juste passé le premier questionnaire d'auto-évaluation.

## 8 Conclusion

Le taux de réussite des étudiants universitaires de première année au cours d'introduction à la programmation n'est pas élevé.

Dans ce cadre, cette recherche vise à identifier des faiblesses dans l'apprentissage de la programmation en incluant le facteur d'auto-évaluation auprès des étudiants inscrits au cours *INFOB131 - Introduction à la programmation* à l'Université de Namur.

Quatre questionnaires (deux questionnaires d'auto-évaluation et deux questionnaires techniques avec degrés de certitude portant sur les mathématiques et sur le langage de programmation python) ont été présentés aux étudiants début décembre en signalant le but formatif et non sommatif de ceux-ci.

29 étudiants ont répondu au premier questionnaire d'auto-évaluation, 17 étudiants ont complété le questionnaire technique de mathématiques et 26 étudiants ont complété le questionnaire technique sur Python. En ce qui concerne le deuxième questionnaire d'auto-évaluation, seulement 3 étudiants l'ont complété.

Début février, un dernier questionnaire post-examen a été soumis afin d'obtenir un retour des étudiants (qu'ils aient ou non complété les premiers questionnaires). Ce dernier questionnaire a été complété par 20 étudiants.

Notre première question de recherche avait pour objectif l'identification de faiblesses dans l'apprentissage de la programmation en utilisant le facteur d'auto-évaluation. Nous avons décelé des points de matière posant problème tant au niveau des mathématiques (logique des propositions, logique des prédicats et calcul booléen) qu'au niveau de l'apprentissage du langage de programmation Python (boucles et fichiers). Par ailleurs, nous avons également pu mettre en avant le fait que la plupart des étudiants ont tendance à surestimer leur niveau de connaissance.

Par notre deuxième question de recherche, nous explorons des pistes d'amélioration dans l'apprentissage de la programmation en utilisant le facteur d'auto-évaluation.

Il serait intéressant de poursuivre cette recherche pour augmenter le nombre de réponses obtenues, mais également en créant et testant les effets de modules d'apprentissage inspirés par les questionnaires auprès des étudiants et des enseignants.

En effet, il est fort probable que de tels modules d'apprentissage, intégrant de l'auto-évaluation, aideraient à la fois les enseignants en permettant d'identifier les faiblesses des étudiants, mais également les étudiants en les aidant à prendre conscience qu'ils surestiment ou sous-estiment leurs connaissances.



## 9 Bibliographie

### Références

- [1] Jens Bennedsen and Michael E. Caspersen. Failure rates in introductory programming. *SIGCSE Bull.*, 39(2) :32–36, jun 2007.
- [2] Christopher Watson and Frederick W.B. Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, page 39–44, New York, NY, USA, 2014. Association for Computing Machinery.
- [3] Jens Bennedsen and Michael E. Caspersen. Failure rates in introductory programming : 12 years later. *ACM Inroads*, 10(2) :30–36, apr 2019.
- [4] Cedric Libert and Wim Vanhoof. Un dispositif d'évaluation continue en introduction à la programmation. In *Une école numérique pour émanciper ? : Colloque scientifique, Actes de la conférence*, pages 16–19, 2018.
- [5] William T. Tarimo, Fatima Abu Deeb, and Timothy J. Hickey. Early detection of at-risk students in CS1 using teachback/spinoza. *Journal of Computing Sciences in Colleges*, 31(6) :105–111, June 2016.
- [6] Soohyun Nam Liao, Daniel Zingaro, Michael A. Laurenzano, William G. Griswold, and Leo Porter. Lightweight, Early Identification of At-Risk CS1 Students. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, ICER '16, pages 123–131, New York, NY, USA, August 2016. Association for Computing Machinery.
- [7] Jonathan P. Munson and Joshua P. Zitovsky. Models for Early Identification of Struggling Novice Programmers. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 699–704, New York, NY, USA, February 2018. Association for Computing Machinery.
- [8] Nuno Gil Fonseca, Luís Macedo, and António José Mendes. Supporting Differentiated Instruction in Programming Courses through Permanent Progress Monitoring. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 209–214, New York, NY, USA, February 2018. Association for Computing Machinery.
- [9] Michael Kuehn, Jared Estad, Jeremy Straub, Thomas Stokke, and Scott Kerlin. An expert system for the prediction of student performance in an initial computer science course. In *2017 IEEE International Conference on Electro Information Technology (EIT)*, pages 1–6, May 2017.
- [10] Varick Erickson. Identifying At-Risk Computer Science Students Early in Semester Utilizing Data-Driven Models. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 865–870, December 2019.
- [11] Ashok Kumar Veerasamy, Daryl D'Souza, Mikko-Ville Apiola, Mikko-Jussi Laakso, and Tapio Salakoski. Using early assessment performance as early warning signs to identify at-risk students in programming courses. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, October 2020.
- [12] Filipe Dwan Pereira, Elaine H. T. Oliveira, David Fernandes, and Alexandra Cristea. Early Performance Prediction for CS1 Course Students using a Combination of Machine Learning and an Evolutionary Algorithm. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, volume 2161-377X, pages 183–184, July 2019.

- [13] Anthony Estey and Yvonne Coady. Can Interaction Patterns with Supplemental Study Tools Predict Outcomes in CS1? In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '16, pages 236–241, New York, NY, USA, July 2016. Association for Computing Machinery.
- [14] Urvashi Desai, Vijayalakshmi Ramasamy, and James D. Kiper. A study on student performance evaluation using discussion board networks. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 500–506, New York, NY, USA, 2020. Association for Computing Machinery.
- [15] José Figueiredo and Francisco García-Peñalvo. Teaching and Learning Tools for Introductory Programming in University Courses. In *2021 International Symposium on Computers in Education (SIIE)*, pages 1–6, September 2021.
- [16] Leo Leppänen, Juho Leinonen, Petri Ihantola, and Arto Hellas. Predicting Academic Success Based on Learning Material Usage. In *Proceedings of the 18th Annual Conference on Information Technology Education*, SIGITE '17, pages 13–18, New York, NY, USA, September 2017. Association for Computing Machinery.
- [17] Simon Liénardy, Laurent Leduc, Dominique Verpoorten, and Benoit Donnet. Café : Automatic correction and feedback of programming challenges for a cs1 course. In *Proceedings of the Twenty-Second Australasian Computing Education Conference*, ACE'20, page 95–104, New York, NY, USA, 2020. Association for Computing Machinery.
- [18] Lloyd Williams, Kimberly J. Titus, and Jason M. Pittman. How Early is Early Enough : Correlating Student Performance with Final Grades. In *Computing Education Practice 2021*, CEP '21, pages 13–16, New York, NY, USA, January 2021. Association for Computing Machinery.
- [19] Vinayak Hegde and Sushma Rao H.S. A Framework to Analyze Performance of Student's in Programming Language Using Educational Data Mining. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–4, December 2017.
- [20] V. K. Anand, S. K. Abdul Rahiman, E. Ben George, and A. S. Huda. Recursive clustering technique for students' performance evaluation in programming courses. In *2018 Majan International Conference (MIC)*, pages 1–5, March 2018.
- [21] Wei Jin, Cynthia L. Johnson, and Sonal Dekhane. A Guided Inquiry Approach for Detecting and Developing Problem-solving Strategies for Novice Programming Students. In *Proceedings of the 2020 ACM Southeast Conference*, ACM SE '20, pages 211–217, New York, NY, USA, April 2020. Association for Computing Machinery.
- [22] Xavier Massart and Marc Romainville. Attributions causales des étudiants en matière de réussite à l'université. *Recherches en éducation*, 37(37), 2019.
- [23] Valérie Wathelet, Matthieu Dontaine, Xavier Massart, Philippe Parmentier, Sandrine Vieillevoye, and Marc Romainville. Exactitude, déterminants, effets et représentations de l'auto-évaluation chez des étudiants de première année universitaire. *Revue internationale de pédagogie de l'enseignement supérieur*, 32(32 (2)), 2016.
- [24] Xavier Massart, Aveline Mazy, Karine Dejean, and Marc Romainville. Les croyances des étudiants sur la nature de leur intelligence : un facteur de participation dans des dispositifs d'aide à la réussite ? *Revue internationale de pédagogie de l'enseignement supérieur*, 38(38 (1)), 2022.
- [25] Sadia Sharmin, Daniel Zingaro, Lisa Zhang, and Clare Brett. Impact of Open-Ended Assignments on Student Self-Efficacy in CS1. In *Proceedings of the ACM Conference on*

- Global Computing Education*, CompEd '19, pages 215–221, New York, NY, USA, May 2019. Association for Computing Machinery.
- [26] Aman Yadav, Chris Mayfield, Sukanya Kannan Moudgalya, Clif Kussmaul, and Helen H. Hu. Collaborative Learning, Self-Efficacy, and Student Performance in CS1 POGIL. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, pages 775–781, New York, NY, USA, March 2021. Association for Computing Machinery.
- [27] Alex Lishinski and Aman Yadav. Self-evaluation Interventions : Impact on Self-efficacy and Performance in Introductory Programming. *ACM Transactions on Computing Education*, 21(3) :23 :1–23 :28, June 2021.
- [28] Grace Ngai, Winnie W.Y. Lau, Stephen C.F. Chan, and Hong-va Leong. On the implementation of self-assessment in an introductory programming course. *SIGCSE Bull.*, 41(4) :85–89, jan 2010.
- [29] Peter Brusilovsky and Sergey Sosnovsky. Individualized exercises for self-assessment of programming knowledge : An evaluation of quizpack. *J. Educ. Resour. Comput.*, 5(3) :6–es, sep 2005.
- [30] Priscilla Lee and Soohyun Nam Liao. Targeting metacognition by incorporating student-reported confidence estimates on self-assessment quizzes. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 431–437, New York, NY, USA, 2021. Association for Computing Machinery.
- [31] Jean-Luc Gilles. Utilisation des degrés de certitude et normes de réalisme en situation d'examen et d'auto-estimation à fapse-ulg : actes du colloque de l'admee-europe 'dix années de travaux de recherche en évaluation', 18-20 septembre 1996. In *Colloque de l'ADMEE-EUROPE 'Dix années de travaux de recherche en évaluation'*, 1996.
- [32] Jean-Luc Gilles. Impact de deux entraînements à l'utilisation des degrés de certitude chez les étudiants de 1ere candidature de la faculté de psychologie et des sciences de l'éducation de l'ulg. In *Stratégies et médias pédagogiques pour l'apprentissage et l'évaluation dans l'enseignement supérieur, Actes du XVème Colloque de l'AIPU*, pages 311–326. Affaires Académiques de l'Université de Liège, 1997.
- [33] Albert Bandura. Self-efficacy : toward a unifying theory of behavioral change. *Psychological review*, 84(2) :191, 1977.
- [34] Candido Cabo. Student Progress in Learning Computer Programming : Insights from Association Analysis. In *2019 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, October 2019.
- [35] Arto Vihavainen. Predicting Students' Performance in an Introductory Programming Course Using Data from Students' Own Programming Process. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 498–499, July 2013.
- [36] Mohammed Hassan and Craig Zilles. Exploring &#x2018;reverse-tracing&#x2019; ; Questions as a Means of Assessing the Tracing Skill on Computer-based CS 1 Exams. In *Proceedings of the 17th ACM Conference on International Computing Education Research*, ICER 2021, pages 115–126, New York, NY, USA, August 2021. Association for Computing Machinery.
- [37] Kim B. Bruce, Robert L. Scot Drysdale, Charles Kelemen, and Allen Tucker. Why math ? *Commun. ACM*, 46(9) :40–44, sep 2003.
- [38] Theresa Beaubouef. Why computer science students need math. *SIGCSE Bull.*, 34(4) :57–59, dec 2002.

- [39] D LECLERCQ and G PLUNUS. Double check, étude de méthodes d'innovation d'évaluations universitaires valides, formatives et efficaces. *Rapport recherche-action, Université de Liège, Service de Technologie de l'Éducation, à paraître, 1996.*
- [40] Jorma Sajaniemi. Guest editor's introduction : Psychology of programming : Looking into programmers heads. *Human technology : an Interdisciplinary Journal on humans in ICT environments, 2008.*
- [41] Sue Jones and Gary Burnett. Spatial ability and learning to program. *Human Technology : An Interdisciplinary Journal on Humans in ICT Environments, 2008.*
- [42] J-M Hoc. La psychologie de la programmation : La percée d'une démarche empirique dans un environnement normatif. *Le travail humain, 46(2) :199–204, 1983.*
- [43] Alan F Blackwell, Marian Petre, and Luke Church. Fifty years of the psychology of programming. *International Journal of Human-Computer Studies, 131 :52–63, 2019.*
- [44] Julie Henry. Mesurer la compréhension en informatique à partir des représentations erronées.
- [45] Amber Settle, John Lalor, and Theresa Steinbach. Reconsidering the impact of cs1 on novice attitudes. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 229–234, 2015.

# Annexes

## A Questionnaire d'auto-évaluation

### Vas-tu réussir ton cours d'introduction à la programmation ?

Ce questionnaire est réalisé dans le cadre d'un mémoire en MASTER 60 HD : **Réussir en informatique : Connaître ses faiblesses pour mieux y répondre.**

Les données collectées dans le cadre de ce mémoire seront anonymisées après traitement.

Ce **premier questionnaire** fait partie d'une liste de 4 questionnaires qui te seront soumis dans le but de t'auto-évaluer sur tes connaissances dans les cours liés à la réussite de ton examen du cours d'introduction à la programmation.

Il sera suivi de deux questionnaires techniques afin de confirmer ton auto-évaluation. Ensuite Le dernier questionnaire te permettra, au besoin, d'adapter ton auto-évaluation du premier questionnaire.

Afin de compléter notre étude, nous te recontacterons après la session de Janvier pour avoir un feedback sur tes résultats d'examen.

paul.peytier@gmail.com [Changer de compte](#)



Brouillon enregistré

\* Indique une question obligatoire

Adresse e-mail \*

paul.peytier@gmail.com

Quel est ton numéro de carte d'étudiant ? \*

TEST

RGPD (Règlement Général sur la Protections des Données) \*

- En soumettant ce formulaire, j'accepte que les informations soient exploitées dans le cadre de la réalisation du mémoire.

Questionnaire d'auto-évaluation

Au cours du questionnaire il va t'être demandé de te positionner sur une échelle de 0 à 10 par rapport à différentes matières, il y aura 10 réponses possibles (excepté pour le niveau d'anglais) :

- 0 : connaissance insuffisante
- 5 : Connaissance moyenne
- 10 : Connaissance parfaite

A combien situerais-tu tes connaissances en Algèbre ? \*

Pour rappel, l'algèbre reprend notamment :

- algèbre des matrices
- déterminant

Algèbre des matrices

$$3 \begin{bmatrix} 3 & -1 & \sqrt{2} \\ \sqrt{3} & 1 & 0 \\ 2 & 3 & -1 \end{bmatrix} - 2 \begin{bmatrix} 4 & -1 & 3\sqrt{2} \\ 2\sqrt{3} & 0 & 1 \\ 3 & -1 & 2 \end{bmatrix}$$

Déterminant d'une matrice

Soit la matrice

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

le déterminant est

$$\det A = a_{11}(-1)^{1+1} \det a_{22} + a_{12}(-1)^{1+2} \det a_{21}$$

0 1 2 3 4 5 6 7 8 9 10

Aucune  
connaissance

Parfaite  
connaissance

A combien situerais-tu tes connaissances en mathématiques discrètes ? \*

Pour rappel, les mathématiques discrètes reprennent notamment :

- logique des propositions et des prédicats
- calcul booléen
- numération

Logique des propositions

$p \vee \neg(p \wedge q)$  Voici la table de vérité :

p	q	$(p \wedge q)$	$\neg(p \wedge q)$	$p \vee \neg(p \wedge q)$
1	1	1	0	1
1	0	0	1	1
0	1	0	1	1
0	0	0	1	1

Logique des prédicats

Les naturels positifs plus petits que 4, sont plus petits que 7.

peut s'écrire  $\forall x \in \mathbb{N} : (x \leq 4) \Rightarrow (x \leq 7)$ ,

ou encore  $\forall x \in \{0, 1, 2, 3, 4\} : (x \leq 7)$ .

Calcul booléen

$$(c + d)(a + b)\overline{c\overline{d}} = (c + d)(a + b)(\overline{c + d})$$

$$= (c + d)(\overline{c + d})(a + b)$$

$$= 0(a + b)$$

$$= 0$$

Numération

Valeur de b	Expression en base b	Forme développée	Expression en base dix
b = 10	1101	$1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$	= 1101
b = 8	$\overline{1101}_8$	$1 \times 8^3 + 1 \times 8^2 + 0 \times 8^1 + 1 \times 8^0$	= 577
b = 2	$\overline{1101}_2$	$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	= 13

0 1 2 3 4 5 6 7 8 9 10

Aucune  
connaissance

Parfaite  
connaissance

## Connaissance de l'anglais

Selon la norme CECRL, quel est ton niveau d'anglais ? \*

- A0 (Aucune notion)
- A1 (Débutant - peut comprendre et utiliser des expressions familières)
- A2 (Peut communiquer lors de tâches simples et habituelles ne demandant qu'un échange d'informations simple et direct sur des sujets familiers et habituels)
- B1 (Intermédiaire - Peut se débrouiller dans la plupart des situations rencontrées en voyage dans une région où la langue cible est parlée)
- B2 (Courant - Peut communiquer avec un degré de spontanéité et d'aisance tel qu'une conversation avec un locuteur natif)
- C1 (Avancé - Peut s'exprimer sur des sujets complexes de façon claire et bien structurée)
- C2 (Bilingue - Peut s'exprimer spontanément, très couramment et de façon précise. Peut rendre distinctes de fines nuances de sens en rapport avec des sujets complexes)

## Connaissance en programmation

Langage

A combien situerais-tu tes connaissances en programmation Python ? \*

0 1 2 3 4 5 6 7 8 9 10

Aucune  
connaissance

Parfaite  
connaissance

Connais-tu un autre langage de programmation que Python ? \*

Votre réponse

---



## B Questionnaire de mathématiques

### Vas-tu réussir ton cours d'introduction à la programmation ?

Ce premier questionnaire technique est réalisé dans le cadre d'un mémoire en MASTER 60 HD : **Réussir en informatique : Connaître ses faiblesses pour mieux y répondre.**

Les données collectées dans le cadre de ce mémoire seront anonymisées après traitement.

Ce **deuxième questionnaire** a pour but d'évaluer tes connaissances en mathématique. Ce questionnaire technique a pour unique vocation de te donner un feedback sur tes faiblesses éventuelles.

Il est conseillé de prendre une feuille de brouillon pour la résolution des différentes questions qui seront posées.

Pour chaque question, 4 réponses te seras proposées et pour chaque question nous te demandons sur une échelle de 0 à 10 ton degré de certitude

Le questionnaire n'a pas de temps de résolution mais celui-ci est estimé à **20 minutes**.

paul.peytier@gmail.com [Changer de compte](#)



\* Indique une question obligatoire

Adresse e-mail \*

paul.peytier@gmail.com

RGPD \*

0 point

En soumettant ce formulaire, j'accepte que les informations soient exploitées dans le cadre de la réalisation du mémoire.

## Mathématique

Cette rubrique se divise en 2 parties, l'une sur l'algèbre l'autre sur les mathématiques discrètes.

Chaque partie comporte 5 questions en vue d'évaluer ton niveau en mathématique.

### Algèbre

- Algèbre des matrices
- Calcule du déterminant

### Mathématiques discrètes

- Logique des propositions
- Logique des prédicats
- Calcul booléen
- Numération
- Codage - décodage

Quel est ton numéro de carte d'étudiant \*

TEST \_\_\_\_\_

Algèbre

Algèbre des matrices

Calcul  $A^2$  et  $A^3$  pour la matrice suivante \*

0 point

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Option 1

$$A^2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad A^3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Option 2

$$A^2 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 2 \end{bmatrix} \quad A^3 = \begin{bmatrix} 8 & 0 & 8 \\ 0 & 0 & 0 \\ 8 & 0 & 8 \end{bmatrix}$$

Option 3

$$A^2 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 2 \end{bmatrix} \quad A^3 = \begin{bmatrix} 4 & 0 & 4 \\ 0 & 0 & 0 \\ 4 & 0 & 4 \end{bmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

Pas du tout sûr    0    1    2    3    4    5    Totalemment sûr

Calcul le produit matriciel AB & BA \*

0 point

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$AB = \begin{bmatrix} 2 & 4 & -3 \\ 1 & 5 & -3 \\ 2 & 8 & -5 \end{bmatrix} \quad BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Option 1

$$AB = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 5 & 8 \\ -3 & -3 & -5 \end{bmatrix} \quad BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Option 2

$$AB = \begin{bmatrix} 3 & 5 & -3 \\ 3 & 4 & -3 \\ 5 & 7 & -5 \end{bmatrix} \quad BA = \begin{bmatrix} 2 & 2 \\ 1 & 0 \end{bmatrix}$$

Option 3

$$AB = \begin{bmatrix} 1 & 5 & -3 \\ 2 & 4 & -3 \\ 3 & 7 & -3 \end{bmatrix} \quad BA = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0    1    2    3    4    5

Pas du tout sûr

Totalement sûr

Calcul l'inverse de la matrice A \*

0 point

$$A = \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix} A^{-1} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 1 & -1 \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$$

Option 1

$$A^{-1} = \begin{bmatrix} -\frac{2}{3} & \frac{2}{3} \\ \frac{1}{3} & -1 \end{bmatrix}$$

Option 2

$$A^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{-1}{2} \\ -\frac{1}{4} & \frac{3}{4} \end{bmatrix}$$

Option 3

$$A^{-1} = \begin{bmatrix} \frac{2}{3} & \frac{-2}{3} \\ -\frac{1}{3} & 1 \end{bmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Calcule  $D = 3A(B + 2C)$  \*

0 point

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & 8 \end{bmatrix} B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ -1 & 9 \end{bmatrix} C = \begin{bmatrix} -1 & 0 \\ 0 & -2 \\ 1 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 6 & 18 \\ 12 & 243 \end{bmatrix}$$

Option 1

$$D = \begin{bmatrix} 90 & 73 \\ 261 & 219 \end{bmatrix}$$

Option 2

$$D = \begin{bmatrix} 6 & 87 \\ 12 & 168 \end{bmatrix}$$

Option 3

$$D = \begin{bmatrix} 6 & 87 \\ 12 & 313 \end{bmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

## Algèbre

Calcule du déterminant de matrice

Calculer le déterminant de la matrice A \*

0 point

$$A = \begin{bmatrix} m & 2 & -2 \\ 0 & m & 1 \\ m & 1 & 0 \end{bmatrix}$$

$$\det = -2m^2 + 3m$$

Option 1

$$\det = 2m^2 + m$$

Option 2

$$\det = -2m^2$$

Option 3

$$\det = 2m^2$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

## Mathématique discrètes

### Logique des propositions

Traduisez en logique des propositions les affirmations suivantes à l'aide de :  $p$ ,  $q$  et  $r$

- $p$  = Le marché boursier augmente.
- $q$  = Les taux d'intérêts augmentent.
- $r$  = Il y aura une récession

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr

Le marché boursier est en train d'augmenter, et si les taux d'intérêts viennent à augmenter, alors il y aura une récession

\* 0 point

- $p \wedge (q \vee r)$
- $(p \wedge q) \vee r$
- $p \wedge (q \Rightarrow r)$
- $(p \wedge q) \Rightarrow r$

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr



Quelle est la contraposée de la proposition suivante : \*

0 point

*Les fleurs de coquelicot sont rouges*

- Les fleurs de coquelicot ne sont pas rouges
- Ce qui n'est pas rouge ne peut être une fleur de coquelicot
- Ce qui est rouge est une fleur de coquelicot
- Ce qui n'est pas rouge est une fleur de coquelicot

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Laquelle de ses propositions est une tautologie (plusieurs réponses possibles)

\* 0 point

$$((P \wedge Q) \vee R) \Leftrightarrow (P \wedge (Q \vee R))$$

Option 1

$$(p \wedge (p \Rightarrow q)) \Rightarrow p$$

Option 2

$$(p \Rightarrow q) \wedge (-q \Rightarrow -p)$$

Option 3

$$p \vee -(p \wedge q)$$

Option 4

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr

Traduis la proposition suivante en logique propositionnelle

\* 0 point

*Elle partira au marché à 09h00 ou 10h00 et si elle part à 09h00, elle finira ses courses à 11h00*

**p** = Elle partira au marché à 09h00

**q** = Elle partira au marché à 10h00

**r** = elle finira ses courses à 11h00

$$\neg p \Rightarrow q \vee r$$

Option 1

$$(p \Rightarrow \neg q) \wedge (p \vee r)$$

Option 2

$$(p \vee q) \wedge (p \Rightarrow r)$$

Option 3

$$p \vee q \Rightarrow r$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Représenter, grâce à la logique du premier ordre, la proposition suivante

Considère la proposition suivante et nie là \*

0 point

$$\forall x \in \mathbb{R}^+ \forall z \in \mathbb{R} \exists y \in \mathbb{R}^+ : xy > z$$

$$\exists x \in \mathbb{R}^+ \exists z \in \mathbb{R} \forall y \in \mathbb{R}^+ : xy \leq z$$

Option 1

$$\forall x \in \mathbb{R}^+ \forall z \in \mathbb{R} \exists y \in \mathbb{R}^+ : xy \leq z$$

Option 2

$$\exists x \in \mathbb{R}^+ \exists z \in \mathbb{R} \exists y \in \mathbb{R}^+ : xy \leq z$$

Option 3

$$\forall x \in \mathbb{R}^+ \exists z \in \mathbb{R} \forall y \in \mathbb{R}^+ : xy \leq z$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Soit les ensembles suivant

- E un ensemble d'étudiants
- P un ensemble de professeurs
- C un ensemble de cours (matières) différents

Partons du postulat que chaque cours peut être enseigné par tous les professeurs.

Soit le prédicat  $A(e,p,c)$  signifiant : l'étudiant  $e$  appartenant à l'ensemble  $E$  aime le cours  $c$  appartenant à l'ensemble  $C$  qui est donné par le professeur  $p$  appartenant à l'ensemble  $P$ .

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr

Traduit en français la phrase suivante \*

0 point

$$\exists p \in P, \forall e \in E, \forall c \in C : \neg A(e, p, c)$$

- Un professeur au moins donne un cours qu'aucun étudiants aiment.
- Un professeur au moins ne donne aucun cours que tous les étudiants aiment.
- Un professeur au moins donne un cours que tous les étudiants aiment.
- Un professeur au moins ne donne plusieurs cours que les étudiants aiment

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr

Considère trois entiers naturels  $x$ ,  $y$  et  $z$ . \*

0 point

Soit la proposition  $p$  suivante :

$\text{Si } x = 3 \text{ alors } y = 5 \text{ et } z = 1$

Quelle affirmation est correct ?

- P est équivalent à : Si  $y = 5$  et  $z = 1$  alors  $x = 3$
- P est équivalent à : Si  $y \neq 5$  et  $z = 1$  alors  $x \neq 3$
- P est équivalent à : Si  $x \neq 3$  alors  $y \neq 5$  ou  $z \neq 1$
- $\neg P$  est : Si  $x = 3$  et  $y \neq 5$  et  $z \neq 1$
- $\neg P$  est : Si  $x = 3$  alors  $y \neq 5$  ou  $z \neq 1$

Quel est ton degré de certitude de la question précédente \*

0      1      2      3      4      5

Pas du tout sûr                            Totalemement sûr

Pour la proposition suivant, Anne affirme quelle est fausse et Jean prend la valeur  $x=5$  pour prouver qu'elle a tort, qui à raison ? \*

0 point

$\exists x \in \mathbb{R} : x < 3 \Rightarrow x > 4$

- Jean a raison car le conséquent ( $5 < 3$ ) est faux et l'antécédent ( $5 > 4$ ) est vraie, la proposition est vraie
- Jean a tort car l'antécédent ( $5 < 3$ ) est faux et le conséquent ( $5 > 4$ ) est vraie, la proposition est fausse
- Jean a raison car l'antécédent ( $5 < 3$ ) est faux et le conséquent ( $5 > 4$ ) est vraie, la proposition est vraie
- Jean a tort car le conséquent ( $5 < 3$ ) est faux et l'antécédent ( $5 > 4$ ) est vraie, la proposition est vraie

Quel est ton degré de certitude de la question précédente \*

0      1      2      3      4      5

Pas du tout sûr                            Totalemement sûr

Donne la forme normale conjonctive la plus simple possible équivalente à \* 0 point  
cette fonction

$$f(a, b, c) = a\bar{b} + \bar{b}c + ac + ab\bar{c} + \bar{a}b\bar{c} + \bar{a}bc$$

$$\bar{a}b\bar{c}$$

Option 2

$$\bar{a}b + \bar{c}$$

Option 3

$$a + \bar{b} + c$$

Option 4

$$\bar{\bar{a}b\bar{c}}$$

Option 1

Quel est ton degré de certitude de la question précédente \*

0    1    2    3    4    5

Pas du tout sûr                            Totalemment sûr

A partir du diagramme de Karnaugh, donne la forme normale disjonctive la \* 0 point plus simple possible

	$ab$	$a\bar{b}$	$\bar{a}b$	$\bar{a}\bar{b}$
$c$	1	1	1	1
$\bar{c}$	1	1	1	

$$a + \bar{a}bc + \bar{b}$$

Option 1

$$a + \bar{b} + c$$

Option 2

$$\bar{a}\bar{b}\bar{c}$$

Option 3

$$\bar{a} + \bar{b} + \bar{c}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalemment sûr



Simplifie l'expression booléenne suivante \*

0 point

$$(p + r)\overline{q}r(p + \overline{q})$$

$$\overline{p}\overline{q} + p\overline{q}$$

Option 1

$$p\overline{q}$$

Option 2

$$p\overline{r} + \overline{q}r$$

Option 3

$$p\overline{q} + \overline{q}r$$

Option 4

Quel est ton degré de certitude de la question précédente \*

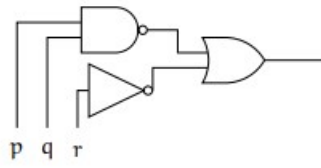
0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

A partir du schéma logique suivant, déduis l'expression booléenne \*

0 point



$$\overline{pq} + \bar{r}$$

Option 1

$$\overline{pqr}$$

Option 2

$$\overline{pq} + r$$

Option 3

$$pq + \bar{r}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

## Mathématiques discrètes

### Système de numération et de représentation

742 est en base 10, donne le résultat en base 3. \*

0 point

- 110011
- 1000111
- 1110001
- 1001221

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

Quel est le résultat en base 12 de l'opération suivante en base 12 :  $100 - 1$  ? \*

0 point

- BB
- B1
- 84
- 48

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

1010101 est en base 2, quel est le résultat en base 8 ? \*

0 point

- 105
- 125
- 131
- 85

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

Quel est le résultat en base 5 de 1A53 base 13 ? \*

0 point

- 111310
- 111302
- 111312
- 111303

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

Soit la notation en virgule flottante. Quelle est la valeur en base 10 du réel \* 0 point suivant  
{ [ 00124 ], e = 3, b = 4, s = -1 } ?

- 1.75
- 1.75
- 0.75
- 0.75

Quel est ton degré de certitude de la question précédente \*

- |                 |                       |                       |                       |                       |                       |                       |                |
|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|                 | 0                     | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Pas du tout sûr | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Totalement sûr |

## Système de numération et de représentation

### Codage-Décodage

Donne la matrice génératrice de ce code \*

0 point

Soit un codage  $\varnothing : B^3 \rightarrow B^6$  avec les mots-code suivants : 101111 110011 111101

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Option 1

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Option 2

$$G = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Option 3

$$G = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

Pas du tout sûr    0    1    2    3    4    5    Totalemment sûr

Donne la liste des syndromes de l'exercice précédent, ainsi que la matrice de contrôle sur base de la matrice génératrice. \* 0 point

$$H' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ Syndrome} = \begin{cases} 000 = 000000 .H' \\ 001 = 000001 .H' \\ 010 = 000010 .H' \\ 011 = 000011 .H' \\ 100 = 000100 .H' \\ 101 = 000101 .H' \\ 110 = 001000 .H' \\ 111 = 001001 .H' \end{cases}$$

Option 1

$$H' = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ Syndrome} = \begin{cases} 000 = 000000 .H' \\ 001 = 000001 .H' \\ 010 = 000010 .H' \\ 011 = 100000 .H' \\ 100 = 000100 .H' \\ 101 = 000101 .H' \\ 110 = 010000 .H' \\ 111 = 010001 .H' \end{cases}$$

Option 2

$$H' = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ Syndrome} = \begin{cases} 000 = 000000 .H' \\ 001 = 000100 .H' \\ 010 = 000010 .H' \\ 011 = 001000 .H' \\ 100 = 000001 .H' \\ 101 = 100000 .H' \\ 110 = 010000 .H' \\ 111 = 010100 .H' \end{cases}$$

Option 3

$$H' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ Syndrome} = \begin{cases} 000 = 000000 .H' \\ 001 = 000100 .H' \\ 010 = 000010 .H' \\ 011 = 000110 .H' \\ 100 = 000001 .H' \\ 101 = 000101 .H' \\ 110 = 000011 .H' \\ 111 = 000111 .H' \end{cases}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0      1      2      3      4      5

Pas du tout sûr                            Totalemnt sûr

Supposons la liste des syndromes suivant, Donne la matrice H correspondante.

\* 0 point

Syndrome	Correction
000	00000
001	00001
010	00010
100	00100
011	01000
101	10000
110	00110
111	01100

$$H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Option 1

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Option 2

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Option 3

$$H = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0      1      2      3      4      5

Pas du tout sûr                            Totalemment sûr



Nous travaillons avec la fonction de codage systématique suivant pour laquelle tu dois déterminer la matrice génératrice.

\* 0 point

$\phi : \mathbb{B}^3 \rightarrow \mathbb{B}^5$  telle que nous obtenons les mots – code suivants :

00000  
00110  
01001  
01111  
10010  
10100  
11011  
11101

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Option 1

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Option 2

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Option 3

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Option 4

Quel est ton degré de certitude de la question précédente \*

0    1    2    3    4    5

Pas du tout sûr                            Totalemment sûr

## C Questionnaire de programmation en langage Python

### Vas-tu réussir ton cours d'introduction à la programmation ?

Ce deuxième questionnaire technique est réalisé dans le cadre d'un mémoire en MASTER 60 HD : **Réussir en informatique : Connaître ses faiblesses pour mieux y répondre.**

Les données collectées dans le cadre de ce mémoire seront anonymisées après traitement.

Ce **troisième questionnaire** a pour but d'évaluer tes connaissances dans le cours d'introduction à la programmation.

Ce questionnaire technique a pour unique vocation de te donner un feedback sur tes faiblesses éventuelles.

Il est conseillé de prendre une feuille de brouillon pour la résolution des différentes questions qui seront posées.

Le questionnaire n'a pas de temps de résolution mais celui-ci est estimé à **20 minutes**.

paul.peytier@gmail.com [Changer de compte](#)



Brouillon enregistré

\* Indique une question obligatoire

Adresse e-mail \*

paul.peytier@gmail.com

Quel est ton numéro de carte d'étudiant ? \*

TEST

RGPD \*

0 point



En soumettant ce formulaire, j'accepte que les informations soient exploitées dans le cadre de la réalisation du mémoire.

## Questionnaire technique - Les variables

Quelle sera la valeur de la variable "x" à la fin de l'exécution de ce code ? \* 0 point

```
1 s = "1"  
2 b = 2  
3 x = s  
4 x += "2"  
5  
6 print(x)
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Quelle sera la valeur de la variable "a" à la fin de l'exécution de ce code ? \* 0 point

```
1 a = 1  
2 b = -1  
3 a = a * b  
4 a = a + b  
5  
6 print(a)
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

## Questionnaire technique - Les conditions

Quelle est la différence entre ces 2 morceaux de code ? \*

0 point

```
1 # code 1
2
3 if maVariable == 5:
4     print("Bonjour")
5 else :
6     print("Au revoir")
7
8 # code 2
9
10 if maVariable == 5:
11     print("Bonjour")
12 print("Au revoir")
```

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 s = 2
2 t = 5
3
4 if s != 2 and t == 5:
5     print("Bonjour")
6 elif s == 2 or t == 4:
7     print("Salut")
8 else :
9     if s == 2 and t == 5:
10        print("Parfait")
11    else :
12        print("Au revoir")
```

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr        Totalemement sûr

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 a = 2
2 b = 3
3 w = 17
4 x = 11
5
6 if a == 1:
7     if b == 3:
8         print("Je suis")
9     else:
10        print("Je ne suis pas")
11
12    if w == 17:
13        print("Genial")
14    elif x == 11:
15        print("Heureux")
16 else:
17    print("Rien")
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr        Totalemement sûr

## Questionnaire technique - Les fonctions

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 def myFunction(fname, lname):  
2     print(fname + " " + lname)  
3  
4 myFunction("Jean")  
5 myFunction("Denis")
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 def my_function(child4, child3, child2, child1):  
2     print("The youngest child is " + child3)  
3  
4 my_function(child1 = "Emil", child2= "Tobias", child3="Linus", child4="Andrew")
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Quel sera le résultat pour la variable "b" après l'exécution de ce programme \* 0 point ?

```
1 a = 5
2
3 def f(a):
4     return a + 1
5
6 def g(a):
7     return a * 10
8
9 def h(a):
10    a = a * 2
11    return f(a) + g(a)
12
13 a = h(a)
14 b = a
15 print(b)
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

	0	1	2	3	4	5	
Pas du tout sûr	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalement sûr

## Questionnaire technique - Les boucles

Choisi le(s) bon(s) code(s) qui permettrai(en)t d'afficher toutes les valeurs \* 0 point  
qui sont multiples de 2

```
1 for i in range(10):  
2     if i % 2 == 0:  
3         print(i)
```

Option 1

```
1 for i in range(10):  
2     if i / 2 == 0:  
3         print(i)
```

Option 2

```
1 i = 0  
2 while i < 10:  
3     if i % 2 == 0:  
4         print(i)  
5     i = i + 1
```

Option 3

```
1 while i < 10:  
2     if i / 2 == 0:  
3         print(i)  
4     i = i + 1
```

Option 4

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr



Pour cet exercice, il t'est demandé de calculer la somme des 5 premiers entiers (1,2,3,4,5) dont le résultat sera affecté à une variable nommée **result** à l'aide d'une boucle **for**. \* 0 point

Bien que l'indentation est importante, elle ne sera pas prise en compte pour cet exercice.

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr       Totalement sûr

On veut connaître à partir de quel entier **n** la somme des **n** premiers entiers dépasse 5000. \* 0 point

Pour ce faire, définis une fonction `sum()` qui ne prend rien en argument et trouve un moyen algorithmique pour ce problème.

Par exemple :

Nous voulons connaître la valeur de l'entier **n** dont la somme des **n** premiers entiers dépasse 10.  
Donc on aura :  $1+2+3 = 9$  si on addition la valeur **n**-suivant (4) cela donnera 13, donc on dépasse la valeur 10 à partir de la valeur 4  
Nous aurons en sortie ceci : 4

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr       Totalement sûr

## Questionnaire technique - Les structures de données

Laquelle de ces réponses correspond à une déclaration d'un TUPLE ? \*

0 point

- ("apple","banana","cherry")
- ["apple", "banana", "cherry"]
- {"name": "apple", "color": "green"}
- {"apple", "banana", "cherry"}

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

Supposons une liste de string

\* 0 point

**thislist = ["apple", "banana", "cherry"]**

Comment pourrais-tu vérifier que le mot "apple" est bien présent dans la liste ?

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

- 0      1      2      3      4      5
- Pas du tout sûr                            Totalemment sûr

Supposons le morceau de code suivant, quelle sera sa sortie ? \*

0 point

```
1  thisList = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
2  print(thisList[-4:-1])
```

Votre réponse

---

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 list = [17,12,54,84,79]
2 newList = [1,2,3]
3
4 begin = 3
5 end = 4
6
7 def replace(list , begin , end , newList):
8     maList = list [0:begin]
9     for i in newList:
10        maList.append(i)
11    for i in list [end:]:
12        maList.append(i)
13
14    return maList
15
16 print (replace (list , begin , end , newList))
```

- [17, 12, 54, 1, 2, 3, 84, 79]
- [17, 12, 54, 1, 2, 3, 79]
- [17, 12, 1, 2, 3, 79]
- [17, 12, 1, 2, 3, 84, 79]

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

Ecris ta réponse sur une seule ligne sans espace.

```
1 l = [7,1,4,6,9,5,0,2,3,8]
2 i = 0
3 while l[i] != 0:
4     i = l[i]
5     print(i)
```

Votre réponse

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

Soit la liste de dictionnaires suivant pour laquelle on veut modifier la valeur 'type' pour la marque BMW car il y a une erreur, le type de cette voiture doit être 'Sportive' et non 'Touring'.

De plus, on voudrait connaître le type de la clé : 'Ch'

Comment ferais tu cela en Python ?

```
1 information_car = [{'Marque': 'VW', 'Model': 'Tiguan', 'Ch': 95, '
2 Type': 'Citadine'},
3 {'Marque': 'BMW', 'Model': '6.35', 'Ch': 130, '
Type': 'Touring'},
{'Marque': 'Lamborghini', 'Model': 'Murcielage', '
Ch': 270, 'Type': 'SuperCar'}]
```

Votre réponse

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr

## Les fichiers

Liste de fonctions utiles :

Liste de fonctions utiles :

- `os.getcwd()` : renvoie le nom du répertoire courant
- `os.chdir(path)` : change le répertoire courant en `path`
- `os.mkdir(path)` : crée un répertoire dont le nom est `path`
- `os.rmdir(path)` : supprime le répertoire `path` (s'il est vide)
- `os.listdir(path)` : renvoie la liste des fichiers/répertoires dans `path`
- `os.rename(source, dest)` : renomme `source` en `dest`
- `os.remove(path)` : supprime le fichier `path`
  
- `os.path.exists(path)` : vérifie si `path` existe
- `os.path.getsize(path)` : donne la taille de `path` en bytes
- `os.path.isfile(path)` : vérifie si `path` est un fichier
- `os.path.isdir(path)` : vérifie si `path` est un répertoire
  
- `fh = open(file_name, mode)` ouvre un fichier dans un mode donné
- `fh.close()` ferme un fichier précédemment ouvert
  
- `lines = fh.readlines()` renvoie une liste de chaîne de caractères correspondant aux lignes du fichier (lit tout le fichier en une fois)
- `fh.write(str)` écrit une chaîne de caractères dans un fichier
  
- `pickle.load(file)` : Renvoie l'objet qui a été enregistré dans le fichier
- `pickle.dump(obj, file)` : Enregistre l'objet 'obj' dans le fichier

Décris, en quelques mots, le comportement du code suivant : \*

0 point

```
1 import os
2 path = './zoo.txt'
3
4 if os.path.isfile(path):
5     print(os.path.isfile(path))
6     data = open(path, 'r')
7     content = data.readlines()
8     data.close()
```

Votre réponse

Quel est ton degré de certitude de la question précédente ? \*

Pas du tout sûr    0    1    2    3    4    5    Totalemment sûr

Soit un fichier cheeses.txt qui contient une liste de 3 fromages, chacun des 3 fromages est écrit sur une ligne. \* 0 point

Il t'est demandé d'ajouter un nouveau fromage dans ce fichier chesses.txt, le nom du fromage sera : "Gorgonzola".

Je ne me souviens plus si le fichier existe ou non, fais en sorte de savoir s'il existe et si c'est le cas effectue ton traitement d'ajout.

Affiche ensuite le contenu du nouveau fichier.

Votre réponse

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5  
Pas du tout sûr       Totalemment sûr

## Questionnaire technique - Pèle mèle

Qu'est-ce qui sera affiché à la fin de l'exécution de ce code ? \*

0 point

```
1 def whatIsTheResult (a,b):
2     if len(b)>1:
3         for a in b:
4             print(a)
5
6 a = 2
7 array = [0,1,2,3,4,5,6,7,8,9,10]
8 whatIsTheResult(a, array)
```

- 2
- 2,3,4,5,6,7,8,9,10
- 0,1,2,3,4,5,6,7,8,9,10
- Rien ne sera affiché car il n'y a pas de "print" dans le main

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr       Totalemement sûr

Ce morceau de code calcule la factorielle d'un nombre.

\* 0 point

Pour rappel, la factorielle de 6 notée  $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ .

pour information, la factorielle de 0 est 1.

Le calcul de la factorielle ne s'effectue que sur des entiers positifs.

Il t'est demandé d'écrire une fonction qui calcule la factorielle d'un entier positif de manière récursive et d'y inclure un traitement pour la factorielle de 0 ainsi que pour les nombres négatifs.

```
1 def notRecursive (n , result):
2     for i in range(1, n+1):
3         result = result * i
4     print(result)
5
6 result = 1
7 n = 6
8 notRecursive(n, result)
```

Votre réponse \_\_\_\_\_

Quel est ton degré de certitude de la question précédente ? \*

0 1 2 3 4 5

Pas du tout sûr

Totalement sûr



## D Questionnaire post-examen

### Comment s'est déroulé ton examen ?

Ça y est ! Les examens sont passés et les résultats sont arrivés !

Tu as complété les questionnaires d'auto évaluation.....ou pas....?

Nous te demandons un petit effort pour avancer dans notre mémoire....et permettre aux étudiants des années futures de bénéficier d'un coup de main pour leur réussite future.

Tu trouveras un petit questionnaire que nous te demandons de compléter de manière franche.

Ce questionnaire ne prendra pas plus de 5 minutes.

paul.peytier@gmail.com [Changer de compte](#)



Brouillon enregistré

\* Indique une question obligatoire

Adresse e-mail \*

paul.peytier@gmail.com

Quel est ton numéro de carte d'étudiant ? \*

TEST

Section sans titre

As-tu complété les questionnaires précédents sur ton Auto-évaluation ainsi que deux questionnaires techniques en Python et Mathématique ? \*

Oui

Non

## D.1 L'étudiant a complété les questionnaires

Tu as complété les questionnaires

As-tu réussi ton examen d'introduction à la programmation ? \*

Oui

Non

Les questionnaires t'ont-ils permis d'identifier des lacunes / de la matière à retravailler ? \*

Oui

Non

Que conseillerais/souhaiterais tu voir dans un module d'aide à la réussite ? \*

Votre réponse

---

## D.2 L'étudiant n'a pas complété les questionnaires

Tu n'as pas complété les questionnaires

As-tu réussi ton examen d'introduction à la programmation ? \*

Oui

Non

Pour quelle raison n'as-tu pas complété les questionnaires ? \*

- Je n'ai pas eu le temps
- Je n'ai pas vu qu'il y avait des questionnaires
- Je n'étais pas prêt à réaliser les questionnaires (pas assez de connaissances)
- Je n'avais pas besoin de réaliser les questionnaires (maîtrise de la matière)
- Autre : \_\_\_\_\_

Penses-tu que tu aurais pu identifier des lacunes en t'auto évaluant avec les questionnaires ? \*

- Oui
- Non

## E Population qui a participé à l'étude

TABLE 20: Tableau listant les étudiants ayant participé à l'étude ; chaque étudiant a été représenté par un identifiant ID et le X dans la colonne indique que l'étudiant a participé au questionnaire repris en en-tête de colonne

ID	Math	Python	Auto-éval.1	Auto-éval.2	Post-examen
ID-01	X	X	X		
ID-02	X	X			
ID-03	X	X	X	X	X
ID-04	X	X	X	X	X
ID-05	X	X	X		
ID-06	X	X	X		
ID-07	X	X	X		X
ID-08	X	X	X	X	
ID-09	X	X	X		
ID-10	X	X	X		
ID-11	X	X			
ID-12	X	X	X		
ID-13	X	X	X		
ID-14	X	X	X		
ID-15	X		X		
ID-16	X		X		
ID-17	X				
ID-18		X	X		
ID-19		X			
ID-20		X	X		X
ID-21		X			X
ID-22		X	X		
ID-23		X			
ID-24		X	X		
ID-25		X	X		
ID-26		X	X		
ID-27		X			
ID-28		X			
ID-29		X			
ID-30			X		
ID-31					X
ID-32			X		
ID-33					X
ID-34					X
ID-35					X
ID-36					X
ID-37			X		
ID-38			X		
ID-39					X

Suite à la page suivante

Table 20 – suite de la page précédente

ID	Math	Python	Auto-éval.1	Auto-éval.2	Post-examen
ID-40					X
ID-41					X
ID-42			X		X
ID-43					X
ID-44					X
ID-45					X
ID-46			X		
ID-47					X
ID-48			X		
ID-49					X
ID-50			X		
ID-51					X
ID-52			X		

## F Réponses des étudiants au premier questionnaire d'auto-évaluation

ID	A combien situerais-tu tes connaissances en Algèbre	A combien situerais-tu tes connaissances en mathématiques discrètes ?	Selon la norme CECRL, quel est ton niveau d'anglais ?	A combien situerais-tu tes connaissances en programmation Python ?	Connais-tu un autre langage de programmation que Python ?
ID-01	7	7	B2	7	NON
ID-03	8	8	B1	8	NON
ID-04	5	5	B1	7	NON
ID-05	0	1	B1	3	OUI
ID-06	2	2	B1	6	NON
ID-07	7	9	B1	5	NON
ID-08	7	3	B2	6	NON
ID-09	9	7	B1	6	NON
ID-10	7	8	B2	7	NON
ID-12	8	6	B1	4	OUI
ID-13	5	4	B2	8	OUI
ID-14	8	7	B2	6	NON
ID-15	7	9	B1	6	OUI
ID-16	7	3	B1	4	OUI
ID-18	9	10	B1	10	OUI
ID-20	6	9	C1	7	OUI
ID-22	2	2	B1	6	NON
ID-24	10	7	B2	6	OUI
ID-25	8	8	A2	3	OUI
ID-26	7	8	B1	9	NON
ID-30	4	5	A1	5	OUI
ID-32	9	7	B2	7	NON
ID-37	6	7	B2	7	NON
ID-38	4	2	B1	6	OUI
ID-42	2	2	B2	4	NON
ID-46	10	3	B2	8	NON
ID-48	10	8	B2	6	NON
ID-50	4	4	B2	5	NON
ID-52	7	7	B2	6	NON

TABLE 21 – Réponses au premier questionnaire d'auto-évaluation

# G Réponses des étudiants au questionnaire de mathématiques

Algèbre des matrices															
ID	Q1 (RE)	Q1-DC	Q1 (BR)	Q2 (RE)	Q2-DC	Q2 (BR)	Q3 (RE)	Q3-DC	Q3 (BR)	Q4 (RE)	Q4-DC	Q4 (BR)	Q5 (RE)	Q5-DC	Q5 (BR)
ID-01	3	5	4	1	5	1	3	5	3	3	3	1	3	5	2
ID-02	4	5	4	1	5	1	3	5	3	1	1	1	2	5	2
ID-03	4	5	4	1	4	1	3	5	3	1	3	1	2	4	2
ID-04	4	5	4	1	5	1	3	5	3	4	2	1	2	4	2
ID-05	2	2	4	1	1	1	2	2	3	3	2	1	1	3	2
ID-06	4	1	4	4	1	1	4	0	3	4	0	1	2	2	2
ID-07	4	5	4	1	5	1	3	5	3	4	1	1	2	5	2
ID-08	4	5	4	1	5	1	3	4	3	1	3	1	2	5	2
ID-09	4	5	4	1	5	1	3	5	3	4	0	1	2	5	2
ID-10	4	4	4	1	4	1	3	3	3	1	2	1	2	4	2
ID-11	4	5	4	1	5	1	3	5	3	3	2	1	2	2	2
ID-12	4	3	4	2	5	1	4	1	3	4	3	1	2	5	2
ID-13	1	3	4	4	2	1	4	2	3	2	2	1	2	2	2
ID-14	4	5	4	1	4	1	3	4	3	3	2	1	1	3	2
ID-15	4	5	4	1	5	1	3	5	3	1	5	1	3	0	2
ID-16	2	3	4	1	2	1	3	0	3	1	4	1	2	4	2
ID-17	4	5	4	1	5	1	3	3	3	1	4	1	2	5	2

TABLE 22 – Questions portant sur l’algèbre des matrices - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse

Mathématiques discrètes															
Logique des propositions															
ID	Q6 (RE)	Q6-DC	Q6 (BR)	Q7 (RE)	Q7-DC	Q7 (BR)	Q8-1 (RE)	Q8-1-DC	Q8-1 (BR)	Q8-2 (RE)	Q8-2-DC	Q8-2 (BR)	Q9 (RE)	Q9-DC	Q9 (BR)
ID-01	3	5	4	1	5	2	2	4	2	3	4	4	3	5	3
ID-02	4	5	4	2	5	2	2	4	2	3	4	4	3	5	3
ID-03	3	5	4	2	5	2	0	4	2	4	4	4	3	4	3
ID-04	3	3	4	2	4	2	2	1	2	1	1	4	3	3	3
ID-05	4	3	4	4	3	2	3	2	2	4	2	4	3	3	3
ID-06	3	4	4	2	2	2	1	2	2	0	2	4	3	3	3
ID-07	3	5	4	2	5	2	1	5	2	0	5	4	3	5	3
ID-08	4	3	4	2	4	2	1	2	2	3	2	4	3	3	3
ID-09	3	4	4	4	3	2	2	4	2	4	4	4	3	3	3
ID-10	3	5	4	2	5	2	2	4	2	0	4	4	3	4	3
ID-11	3	5	4	2	5	2	2	2	2	4	2	4	3	2	3
ID-12	4	5	4	1	3	2	2	2	2	3	2	4	3	4	3
ID-13	4	5	4	4	3	2	2	4	2	0	4	4	3	5	3
ID-14	3	3	4	2	4	2	2	4	2	0	4	4	3	5	3
ID-15	3	5	4	2	3	2	1	4	2	3	4	4	3	5	3
ID-16	4	3	4	1	3	2	1	1	2	0	1	4	3	3	3
ID-17	4	5	4	2	5	2	0	5	2	4	5	4	3	5	3

TABLE 23 – Questions portant sur la logique des propositions - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse

Mathématiques discrètes												
Logique des prédicats												
ID	Q10 (RE)	Q10-DC	Q10 (BR)	Q11 (RE)	Q11-DC	Q11 (BR)	Q12 (RE)	Q12-DC	Q12 (BR)	Q13 (RE)	Q13-DC	Q13 (BR)
ID-01	1	5	1	1	4	2	2	4	2	3	5	3
ID-02	1	5	1	4	3	2	5	0	2	3	5	3
ID-03	1	5	1	1	5	2	2	4	2	3	5	3
ID-04	1	5	1	2	3	2	4	5	2	3	5	3
ID-05	4	1	1	2	2	2	3	4	2	1	2	3
ID-06	2	1	1	1	1	2	2	0	2	1	3	3
ID-07	1	5	1	2	4	2	2	4	2	3	5	3
ID-08	2	0	1	2	0	2	1	0	2	2	0	3
ID-09	1	5	1	2	2	2	2	0	2	3	4	3
ID-10	1	4	1	1	3	2	5	4	2	3	4	3
ID-11	1	5	1	1	2	2	3	5	2	3	5	3
ID-12	2	2	1	3	1	2	3	1	2	4	2	3
ID-13	1	5	1	2	4	2	5	5	2	4	5	3
ID-14	1	3	1	1	2	2	5	0	2	2	3	3
ID-15	1	4	1	2	2	2	5	3	2	3	5	3
ID-16	1	3	1	2	3	2	1	0	2	1	2	3
ID-17	1	5	1	3	5	2	4	5	2	3	5	3

TABLE 24 – Questions portant sur la logique des prédicats - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse

Mathématiques discrètes												
Calcul booléen												
ID	Q14 (RE)	Q14-DC	Q14 (BR)	Q15 (RE)	Q15-DC	Q15 (BR)	Q16 (RE)	Q16-DC	Q16 (BR)	Q17 (RE)	Q17-DC	Q17 (BR)
ID-01	1	0	1	2	0	2	3	0	3	1	4	1
ID-02	4	5	1	2	5	2	2	0	3	1	5	1
ID-03	4	4	1	2	5	2	3	4	3	1	5	1
ID-04	4	5	1	2	5	2	2	0	3	1	5	1
ID-05	1	2	1	4	3	2	3	3	3	2	5	1
ID-06	2	2	1	4	1	2	1	2	3	3	2	1
ID-07	2	5	1	2	5	2	4	4	3	1	5	1
ID-08	4	0	1	2	2	2	4	0	3	1	3	1
ID-09	1	4	1	2	4	2	3	3	3	1	3	1
ID-10	4	5	1	2	4	2	4	2	3	1	4	1
ID-11	2	1	1	2	5	2	4	0	3	1	5	1
ID-12	4	2	1	2	3	2	4	2	3	1	5	1
ID-13	1	1	1	4	2	2	2	2	3	1	4	1
ID-14	2	3	1	1	3	2	3	0	3	1	3	1
ID-15	3	1	1	2	4	2	3	0	3	1	4	1
ID-16	1	0	1	3	0	2	1	0	3	4	3	1
ID-17	4	5	1	2	5	2	2	5	3	1	5	1

TABLE 25 – Questions portant sur le calcul booléen - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse

Mathématiques discrètes															
Systèmes de numération et de représentation															
ID	Q18 (RE)	Q18-DC	Q18 (BR)	Q19 (RE)	Q19-DC	Q19 (BR)	Q20 (RE)	Q20-DC	Q20 (BR)	Q21 (RE)	Q21-DC	Q21 (BR)	Q22 (RE)	Q22-DC	Q22 (BR)
ID-01	4	0	2	2	0	1	3	0	2	2	0	1	1	0	2
ID-02	2	5	2	1	5	1	2	5	2	1	0	1	2	5	2
ID-03	2	4	2	1	5	1	2	5	2	1	5	1	2	4	2
ID-04	2	5	2	4	0	1	2	5	2	1	2	1	2	4	2
ID-15	4	0	2	3	0	1	3	0	2	3	0	1	4	0	2
ID-05	2	5	2	1	5	1	2	5	2	1	5	1	2	1	2
ID-06	2	3	2	4	3	1	2	3	2	1	1	1	2	0	2
ID-07	2	5	2	1	5	1	2	5	2	1	5	1	1	4	2
ID-16	1	0	2	1	0	1	1	0	2	1	0	1	1	0	2
ID-08	3	0	2	2	0	1	1	0	2	3	0	1	1	0	2
ID-09	2	5	2	1	4	1	2	4	2	1	3	1	2	0	2
ID-10	2	5	2	1	4	1	2	5	2	2	0	1	2	0	2
ID-17	2	5	2	1	5	1	2	5	2	1	5	1	3	5	2
ID-11	2	3	2	1	5	1	2	5	2	1	5	1	2	2	2
ID-12	2	5	2	1	5	1	2	5	2	4	5	1	2	2	2
ID-13	2	2	2	3	1	1	2	1	2	2	1	1	2	1	2
ID-14	2	4	2	1	4	1	2	4	2	1	0	1	2	3	2

TABLE 26 – Questions portant sur les systèmes de numération et de représentation - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse

Mathématiques discrètes												
Codage-décodage												
ID	Q23 (RE)	Q23-DC	Q23 (BR)	Q24 (RE)	Q24-DC	Q24 (BR)	Q25 (RE)	Q25-DC	Q25 (BR)	Q26 (RE)	Q26-DC	Q26 (BR)
ID-01	2	3	1	2	0	1	2	4	2	4	1	4
ID-02	1	5	1	1	0	1	1	0	2	4	5	4
ID-03	1	4	1	1	4	1	2	4	2	4	4	4
ID-04	2	0	1	1	0	1	2	0	2	4	5	4
ID-15	1	3	1	3	1	1	3	2	2	3	0	4
ID-05	4	2	1	3	1	1	2	1	2	3	2	4
ID-06	2	1	1	1	4	1	2	1	2	3	0	4
ID-07	1	5	1	1	5	1	2	4	2	4	5	4
ID-16	1	0	1	1	0	1	1	0	2	1	0	4
ID-08	1	0	1	3	0	1	2	0	2	1	0	4
ID-09	1	5	1	1	4	1	2	5	2	4	5	4
ID-10	1	4	1	1	3	1	3	3	2	4	4	4
ID-17	2	0	1	4	0	1	3	0	2	2	0	4
ID-11	1	5	1	1	5	1	1	5	2	4	5	4
ID-12	1	5	1	3	2	1	2	5	2	4	5	4
ID-13	2	0	1	2	1	1	2	1	2	4	1	4
ID-14	2	0	1	1	0	1	4	0	2	3	0	4

TABLE 27 – Questions portant sur le codage-décodage - RE = Réponse Etudiant - DC = Degré de Certitude - BR = Bonne Réponse



## H Réponses des étudiants au questionnaire de programmation en langage Python

ID	Les variables				Les conditions					
	Q1 - RE	Q1 - DC	Q2 - RE	Q2 - DC	Q3 - RE	Q3 - DC	Q4 - RE	Q4 - DC	Q5 - RE	Q5 - DC
ID-01	1	5	1	5	1	5	1	5	1	5
ID-02	1	5	1	5	0	5	0	3	0	2
ID-03	1	5	1	5	1	5	1	5	1	5
ID-04	1	5	1	5	1	5	1	5	1	5
ID-05	0	4	1	4	1	3	0	5	1	3
ID-06	1	5	1	5	1	5	1	5	1	5
ID-07	1	4	1	5	0	5	1	5	1	5
ID-08	1	4	1	4	1	5	1	3	1	4
ID-09	1	5	1	4	1	5	1	5	1	5
ID-10	1	4	1	4	1	4	1	4	1	5
ID-11	1	5	1	5	1	5	0	3	1	5
ID-12	0	4	1	5	1	5	0	5	1	5
ID-13	1	4	1	5	0	4	1	5	1	5
ID-14	1	4	1	4	1	4	1	4	1	4
ID-18	1	5	1	5	1	5	1	5	1	5
ID-19	1	5	1	5	1	5	1	5	1	5
ID-20	1	5	1	5	1	5	1	5	1	5
ID-21	1	5	1	5	1	5	1	4	1	5
ID-22	0	0	1	5	1	5	1	5	1	5
ID-23	0	2	1	5	1	5	0	3	1	5
ID-24	1	5	1	5	1	5	1	5	1	5
ID-25	1	5	1	5	0	5	0	2	1	5
ID-26	0	4	1	5	1	5	0	5	1	5
ID-27	1	5	1	3	1	5	0	5	1	5
ID-28	1	5	1	5	1	5	1	5	1	5
ID-29	1	5	1	5	0	5	0	5	1	5

TABLE 28 – Questions portant sur les variables et les conditions. RE = Réponse Etudiant (0 si réponse fausse, 1 si réponse correcte). DC = Degré de Certitude

ID	Les fonctions						Les boucles							
	Q6 - RE	Q6 - DC	Q7 - RE	Q7 - DC	Q8 - RE	Q8 - DC	Q9-1 - RE	Q9-1 - DC	Q9-2 - RE	Q9-2 - DC	Q10 - RE	Q10 - DC	Q11 - RE	Q11 - DC
ID-01	1	5	0	5	1	5	1	5	1	5	0	5	0	0
ID-02	1	5	1	5	0	5	1	5	1	5	1	3	0	5
ID-03	1	4	1	2	1	5	1	5	1	5	1	5	1	5
ID-04	1	5	0	4	1	4	1	4	1	4	1	5	1	3
ID-05	0	4	1	4	0	4	1	5	0	5	0	0	0	0
ID-06	1	5	0	4	1	5	1	5	1	5	0	5	1	5
ID-07	1	5	1	4	1	5	1	5	1	5	1	5	0	5
ID-08	1	4	1	3	1	2	1	3	1	3	0	1	0	0
ID-09	1	4	0	2	1	5	1	3	1	3	1	5	1	5
ID-10	0	3	0	4	1	3	1	4	1	4	0	4	1	4
ID-11	1	5	1	5	1	3	1	2	0	2	1	5	0	2
ID-12	0	2	1	5	0	3	1	2	0	2	0	2	0	2
ID-13	1	4	0	4	1	4	1	5	1	5	0	4	0	4
ID-14	0	2	0	2	0	2	1	4	1	4	0	3	0	3
ID-18	1	5	1	5	1	5	1	5	1	5	0	5	1	5
ID-19	1	5	1	5	1	5	1	5	1	5	0	5	1	5
ID-20	1	4	1	5	1	5	1	5	1	5	1	5	1	5
ID-21	1	4	1	3	0	2	1	4	1	4	0	3	0	0
ID-22	0	1	1	2	0	0	1	4	0	4	0	0	0	0
ID-23	0	5	1	5	0	3	0	3	0	3	0	0	0	0
ID-24	1	4	1	5	1	5	1	4	1	4	0	5	0	5
ID-25	1	5	0	5	1	3	1	4	0	4	0	0	0	0
ID-26	1	5	1	5	1	5	1	5	0	5	1	5	0	5
ID-27	1	4	1	4	1	5	1	4	1	4	1	5	0	4
ID-28	1	5	1	5	1	4	1	5	1	5	1	5	0	4
ID-29	1	5	1	5	0	5	1	5	1	5	0	3	0	0

TABLE 29 – Questions portant sur les fonctions et les boucles. RE = Réponse Etudiant (0 si réponse fausse, 1 si réponse correcte). DC = Degré de Certitude

Les structures de données														
ID	Q12 - RE	Q12 - DC	Q13 - RE	Q13 - DC	Q14 - RE	Q14 - DC	Q15 - RE	Q15 - DC	Q16 - RE	Q16 - DC	Q17-1 - RE	Q17-1 - DC	Q17-2 - RE	Q17-2 - DC
ID-01	1	5	1	5	0	5	1	5	1	5	1	4	1	4
ID-02	1	5	1	5	1	2	1	5	1	4	1	3	1	3
ID-03	1	5	1	5	1	4	0	4	1	4	1	3	1	3
ID-04	1	5	1	3	0	3	1	5	0	4	1	2	1	2
ID-05	1	5	0	0	0	3	1	2	0	0	0	0	0	0
ID-06	1	5	1	5	1	4	1	5	0	5	1	5	1	5
ID-07	1	5	1	5	1	5	1	5	0	5	1	2	1	2
ID-08	1	4	1	2	0	3	0	1	0	0	0	2	0	2
ID-09	1	4	1	5	0	3	1	4	1	4	1	3	1	3
ID-10	1	5	1	4	1	4	1	3	1	4	1	2	0	2
ID-11	1	5	1	5	0	4	1	1	0	5	0	3	0	3
ID-12	1	1	0	1	0	1	0	1	0	2	0	1	0	1
ID-13	1	5	1	4	0	4	1	4	0	3	1	2	1	2
ID-14	1	3	1	3	1	3	1	4	0	3	1	3	0	3
ID-18	1	5	1	5	0	5	1	4	1	5	1	5	0	5
ID-19	1	5	1	5	1	5	1	5	1	5	1	5	0	5
ID-20	1	5	1	5	0	5	1	4	1	5	1	5	1	5
ID-21	1	3	1	4	1	4	1	4	0	2	0	3	0	3
ID-22	1	5	0	4	0	5	1	0	0	0	0	0	0	0
ID-23	0	2	1	4	0	3	1	2	0	0	0	0	0	0
ID-24	1	3	0	3	0	3	1	3	0	5	0	0	0	0
ID-25	1	4	0	4	0	2	0	2	0	2	0	1	0	1
ID-26	1	5	1	5	1	5	1	5	1	5	0	4	0	4
ID-27	1	4	1	5	1	5	0	4	0	5	1	5	1	5
ID-28	1	5	1	5	1	4	1	4	0	5	1	5	1	5
ID-29	0	5	1	5	1	4	1	5	0	3	0	0	0	0

TABLE 30 – Questions portant sur les structures de données. RE = Réponse Etudiant (0 si réponse fausse, 1 si réponse correcte). DC = Degré de Certitude

ID	Les fichiers				Pêle-Mêle			
	Q18 - RE	Q18 - DC	Q19 - RE	Q19 - DC	Q20 - RE	Q20 - DC	Q21 - RE	Q21 - DC
ID-01	0	0	0	0	0	3	0	0
ID-02	1	5	0	5	0	0	1	5
ID-03	0	4	0	3	1	5	1	4
ID-04	1	2	0	0	1	3	1	4
ID-05	0	0	0	0	0	3	0	0
ID-06	1	3	0	5	1	3	1	4
ID-07	1	2	1	4	0	3	1	2
ID-08	1	2	0	0	0	2	0	0
ID-09	0	2	1	2	0	2	1	5
ID-10	0	3	0	4	1	5	1	4
ID-11	1	5	0	4	0	0	1	5
ID-12	0	2	0	2	0	1	0	1
ID-13	1	3	0	4	1	2	0	1
ID-14	1	3	1	2	1	4	0	3
ID-18	0	5	0	0	1	5	1	5
ID-19	0	5	1	5	1	5	1	5
ID-20	1	5	1	5	1	5	1	5
ID-21	0	4	1	3	0	2	0	4
ID-22	0	0	0	0	0	5	0	0
ID-23	1	2	0	0	0	4	0	0
ID-24	1	3	0	4	0	4	0	4
ID-25	0	2	0	0	1	4	0	0
ID-26	1	5	0	5	0	3	1	5
ID-27	0	0	0	0	1	4	1	4
ID-28	0	5	0	0	1	4	1	5
ID-29	1	4	0	4	0	3	0	0

TABLE 31 – Questions portant sur les fichiers et questions pêle-mêle. RE = Réponse Etudiant (0 si réponse fausse, 1 si réponse correcte). DC = Degré de Certitude

# I Réponses des étudiants au questionnaire post-examen

ID	As-tu complété les questionnaires précédents sur ton Auto-évaluation ainsi que deux questionnaires techniques en Python et Mathématique ?	As-tu réussi ton examen d'introduction à la programmation ?	Les questionnaires t'ont-ils permis d'identifier des lacunes / de la matière à retravailler ?	Que conseillerais / souhaiterais tu voir dans un module d'aide à la réussite ?	Pour qu'elle raison n'as-tu pas complété les questionnaires ?	Penses-tu que tu aurais pu identifier des lacunes en t'auto évaluant avec les questionnaires ?
ID-03	OUI	OUI	NON	Ça serait bien de pouvoir tester son code		
ID-04	OUI	OUI	OUI	Des questions types d'examens		
ID-07	OUI	OUI	OUI	Beaucoup plus d'exercices		
ID-20	NON	OUI			Je n'ai pas eu le temps	NON
ID-21	NON	OUI	OUI		Autre : J'ai fait celui de programmation mais pas celui en mathématique, par manque de temps	OUI
ID-31	NON	OUI			Je n'ai pas vu qu'il y avait des questionnaires	OUI
ID-33	OUI	OUI	OUI	Je conseille de faire les test		
ID-34	NON	OUI			Je n'ai pas vu qu'il y avait des questionnaires	NON
ID-35	NON	NON			Je n'ai pas vu qu'il y avait des questionnaires	OUI
ID-36	OUI	OUI	OUI	Mettre des exercices supplémentaires en format papier + leurs solutions		
ID-39	OUI	OUI	OUI	Des explications simplifiés dans les matières où j'ai des lacunes		
ID-40	OUI	OUI	OUI			
ID-41	NON	OUI			Je n'ai pas vu qu'il y avait des questionnaires	NON
ID-42	OUI	OUI	OUI	Les fichiers ainsi que la récursivité		

TABLE 32 – Réponses au questionnaire post-examen - Partie 1/2

ID	As-tu complété les questionnaires précédents sur ton Auto-évaluation ainsi que deux questionnaires techniques en Python et Mathématique ?	As-tu réussi ton examen d'introduction à la programmation ?	Les questionnaires t'ont-ils permis d'identifier des lacunes / de la matière à retravailler ?	Que conseillerais / souhaiterais tu voir dans un module d'aide à la réussite ?	Pour qu'elle raison n'as-tu pas complété les questionnaires ?	Penses-tu que tu aurais pu identifier des lacunes en t'auto évaluant avec les questionnaires ?
ID-43	OUI	OUI	OUI	Une partie sur la réversivité		
ID-44	NON	NON			Je n'ai pas vu qu'il y avait des questionnaires	OUI
ID-45	NON	OUI			Je n'avais pas besoin de réaliser les questionnaires (maitrise de la matière)	OUI
ID-47	NON	OUI			Je n'ai pas vu qu'il y avait des questionnaires	NON
ID-49	NON	OUI			Je n'ai pas vu qu'il y avait des questionnaires	OUI
ID-51	NON	OUI			Je n'avais pas besoin de réaliser les questionnaires (maitrise de la matière)	NON

TABLE 33 – Réponses au questionnaire post-examen - Partie 2/2