

# Expérimentation du test PIKA dans les universités françaises en 2025 : un premier retour d'expérience

Sébastien Hoarau<sup>1</sup>, Sondre Bolland<sup>2</sup>, Antoine Meyer<sup>3</sup>, Martin Rosalie<sup>4</sup>,  
Vincent Boudet<sup>5</sup>, François Puitg<sup>6</sup>, and Christophe Declercq<sup>1</sup>

<sup>1</sup> LIM, Université de La Réunion

<sup>2</sup> Department of Informatics, University of Bergen

<sup>3</sup> Univ Gustave Eiffel, CNRS, LIGM, F-77454 Marne-la-Vallée, France

<sup>4</sup> LGDP, UMR 5096, Université de Perpignan Via Domitia, CNRS, Perpignan, France

<sup>5</sup> LIRMM, Univ. Montpellier, CNRS

<sup>6</sup> LIG, Université Grenoble Alpes

**Résumé** Cette communication relate l'expérimentation du test PIKA de connaissances et compétences en programmation auprès des étudiants de douze universités françaises à la rentrée 2025. Le test PIKA, élaboré à l'Université de Bergen en 2023 et ajusté chaque année, a été diffusé auprès de onze universités nordiques en 2024 et 2025. Il a pour objectif de mesurer l'influence de l'introduction d'un enseignement de l'informatique au niveau secondaire en Norvège en 2020, sur les connaissances et compétences en programmation des étudiants entrant à l'université depuis 2023. L'expérimentation menée en France a concerné 1025 étudiants de première année de licence informatique ou d'un portail y menant et montre des résultats au test PIKA significativement meilleurs pour les étudiants ayant suivi la spécialité NSI en classe de première. Les analyses globales permettent de discuter les compétences les mieux maîtrisées ainsi que la perception du test par les étudiants. Une analyse détaillée par université permet de discuter les différences de résultats en fonction du mode d'administration du test et des spécificités des publics accueillis dans chaque université. Les perspectives ouvertes vont de la généralisation du test à plus de formations et d'universités, à la réflexion sur l'adaptation des cursus aux nouvelles compétences des étudiants entrants.

## Introduction

Suite aux récentes mises à jour du programme scolaire national norvégien, la pensée informatique et la programmation sont devenues des éléments fondamentaux de l'enseignement primaire et secondaire, entraînant un afflux d'étudiants arrivant dans l'enseignement supérieur avec une expérience préalable en programmation. Cette évolution est susceptible d'influencer l'enseignement de l'informatique à l'université, car des connaissances de base pourraient permettre d'aborder plus tôt des sujets plus avancés. Cependant, la qualité de l'enseignement de la programmation au niveau secondaire étant variable, il est essentiel d'évaluer les connaissances préalables des étudiants. Pour répondre à ce besoin,

des chercheurs de l'Université de Bergen ont conçu le « Test nordique des connaissances préalables en programmation », renommé depuis PIKA (Prior Informatics Knowledge Assessment), afin d'évaluer les connaissances et compétences en programmation des nouveaux étudiants par rapport aux éléments de base du cours d'introduction à la programmation (CS1). Les résultats préliminaires de cette étude ont été publiés en 2024 [1], puis en 2025 [3], montrant une influence très significative de la réforme des programmes sur les connaissances et compétences préalables des étudiants commençant un cursus d'informatique à l'université.

Les connaissances à évaluer pour le test ont été définies par un groupe d'experts des universités norvégiennes selon le protocole décrit dans [2]. Les connaissances évaluées concernent exclusivement la compétence « évaluer », selon la terminologie définie dans le référentiel de compétences en programmation RCP [4], basé sur la description de Selby et Woollard [6] des compétences de la pensée informatique [7]. En effet, la maîtrise de la compétence « évaluer » semble plus facile à estimer par une modalité de type QCM que celle des compétences « modéliser », « anticiper » et « décomposer », qui est susceptible de nécessiter des tâches plus complexes (écriture de spécifications, d'algorithmes, de code, ou conception de l'architecture d'un petit projet par exemple). Savoir « évaluer » un programme semble un pré-requis raisonnable à en écrire, ce qui peut justifier l'usage exclusif de cette compétence dans ce test.

L'enseignement scolaire de l'informatique en France ayant été considérablement renforcé par la création de l'enseignement de spécialité Numérique et Sciences Informatiques (NSI) en classe de première à la rentrée 2019, puis en classe de terminale à la rentrée 2020, il nous a semblé pertinent de chercher à tester si des causes similaires pouvaient engendrer les mêmes effets. Une collaboration a donc été proposée aux responsables de licence informatique de 56 universités françaises en août 2025, dont 12 ont réussi à l'administrer à leurs étudiants dès septembre 2025. Faire passer le test PIKA aux étudiants français a été possible grâce à la proximité des contenus d'enseignement de la programmation au secondaire en Norvège et en France, notamment autour du même langage Python, et à un travail d'internationalisation de la plateforme.

1151 étudiants de 12 universités françaises ont passé le test PIKA entre le 1er et le 21 septembre 2025. Le test comportait outre les 48 questions de programmation, des questions sur le cursus de l'étudiant au lycée et en particulier ses choix de spécialités en informatique, en mathématiques et en sciences et technologies. Seules ont été retenues les réponses des étudiants n'ayant pas encore suivi de cours d'informatique à l'université soit 1025 réponses. La répartition par université n'est pas homogène, chaque établissement ayant été libre de faire passer le test soit en présentiel sur des créneaux réservés, soit en ligne à la discrétion de l'étudiant. Il sera tenu compte de ce choix dans nos analyses.

Nous présentons tout d'abord l'analyse globale des résultats au test PIKA portant sur l'ensemble des établissements participants<sup>7</sup>, puis une analyse détaillée par université.

---

7. Des informations plus détaillées sur le test (connaissances évaluées, exemples de questions, statistiques) sont disponibles à l'adresse : <https://pika-test.org>.

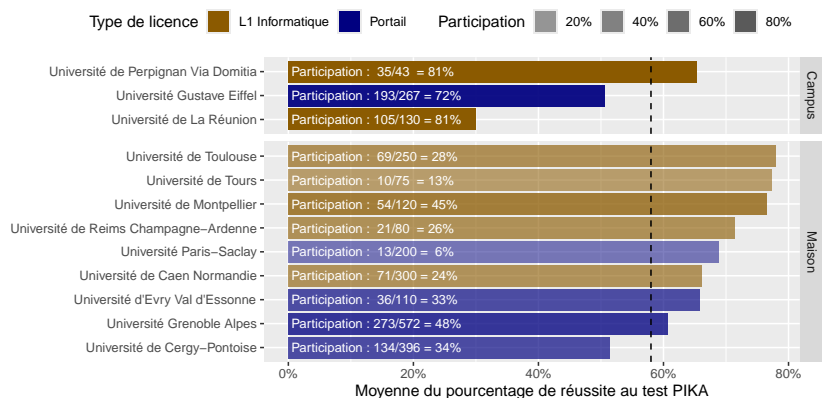


FIGURE 1. Résultats par université (participation en quotient et pourcentage).

### Analyse globale

La figure 1 montre les scores moyens par université pour les 12 établissements ayant fait passer le test à plus de 10 étudiants. Les différences les plus importantes peuvent être expliquées par le mode de passation du test en présentiel pour tous les étudiants pour les trois premiers établissements (Perpignan, Gustave Eiffel et La Réunion), et à distance pour une partie des étudiants pour les autres universités. Un biais de sélection a pu intervenir, les étudiants ayant le plus d'expérience en programmation pouvant être plus motivés pour passer ce test. On constate aussi une différence significative entre les établissements proposant une première année de licence d'informatique (L1) et ceux proposant un portail (première année menant à différentes licences).

Le principal résultat de cette expérimentation concerne les scores des étudiants, selon leur choix de la spécialité NSI en première.

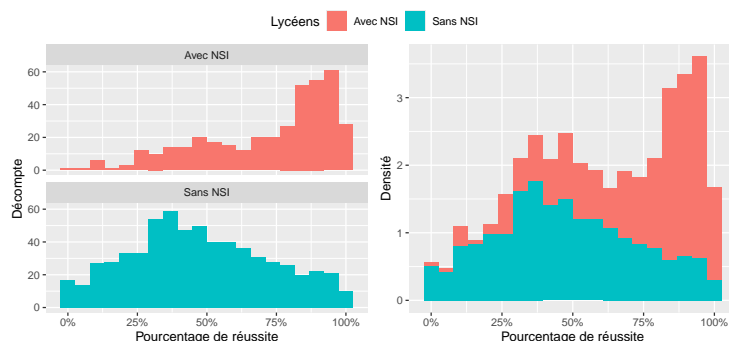


FIGURE 2. Distribution des scores.

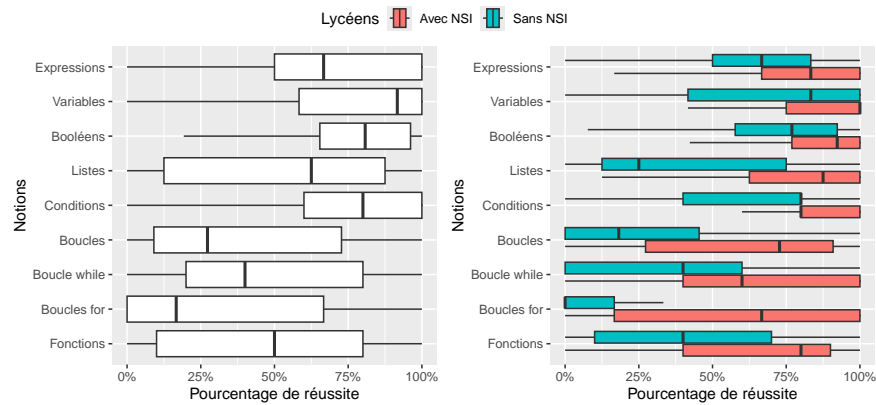


FIGURE 3. Scores par notions.

Sur le graphique de la figure 2, on constate un pic de score autour de 90% pour les étudiants ayant suivi NSI et un pic autour de 40% pour ceux n'ayant pas suivi cet enseignement. Un test de la somme des rangs de Wilcoxon montre une différence très significative (P-value inférieure à 0,01) entre ces deux cohortes.

L'analyse globale par notions (voir figure 3) montre une très bonne maîtrise des premiers concepts - expressions, variables, booléens, conditionnelles - et des scores en moyenne inférieurs pour les listes, les boucles et les fonctions. Le score inférieur pour la boucle `for` par rapport à la boucle `while` peut sembler surprenant, la boucle bornée étant enseignée dès les collèges en France. Le score médiocre aux questions sur la boucle `for` peut en fait s'expliquer par la nature des questions posées, qui portent exclusivement sur des boucles du type `for` e `in` liste alors que c'est principalement la boucle `for i in range(n)` qui est enseignée en première NSI.

La difficulté du test perçue par les étudiants (voir figure 4) montre une majorité relative d'avis « neutres » et en proportions équivalentes des avis soit « facile » ou « très facile » principalement pour les lycéens avec NSI, soit « difficile » ou « très difficile » pour les lycéens sans NSI. Cela semble témoigner d'un test correctement étalonné par rapport au niveau moyen des étudiants.

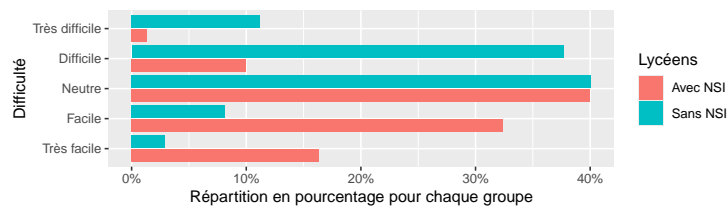


FIGURE 4. Difficulté perçue du test.

Les verbatims laissés par les étudiants en fin de test sont nombreux à faire état d'un test « trop long ». Une analyse plus détaillée des corrélations entre scores aux différentes questions pourrait permettre de supprimer des questions redondantes sans diminuer la couverture des notions de programmation par le test. Cette analyse est en cours sur les résultats au niveau international et a pour objectif de ramener la durée de passation du test de 45 minutes à 30 minutes.

## Analyse détaillée par université

L'analyse des résultats des trois universités ayant administré le test en présentiel (Perpignan, Gustave Eiffel et La Réunion) peut inclure des résultats plus significatifs par rapport au cursus actuel et antérieur des étudiants, l'ensemble d'une cohorte ayant été interrogée. La L1 Informatique de Perpignan ne fait partie d'aucun portail ou regroupement avec un petit effectif (65 étudiants inscrits par an). Le test a été passé uniquement par des primo-entrants. En moyenne, il y a plus d'étudiants ayant choisi l'option NSI (22 étudiants sur 35) qu'au niveau national et, réciproquement, moins d'étudiants ayant choisi la spécialité mathématiques. Ces choix d'options et la petite cohorte expliquent en partie les résultats obtenus (65%) qui sont un peu au-dessus de la moyenne nationale (58%). Les résultats sont significativement meilleurs pour les étudiants ayant suivi NSI comme au niveau national.

L'université de La Réunion se distingue par le score le plus faible (30%), malgré un taux de lycées offrant la spécialité NSI (26 lycées sur 30) très supérieur à celui de l'hexagone. Cette proportion ne se retrouve pas dans les entrants en licence d'informatique qui ne sont que 49 sur 105 à avoir suivi la spécialité NSI en classe de première. Un nombre significatif d'étudiants viennent du baccalauréat technologique ou professionnel. Une majorité (55%) des étudiants ont trouvé le test difficile ou très difficile.

L'université Gustave Eiffel est celle qui fait passer le test en présentiel aux plus grand nombre d'étudiants (193, en excluant les redoublants). Le score global est bon (50,6%) avec une différence significative par rapport à NSI. Les profils d'étudiants sont variés entre portail, double licence, néo-entrants ou réorientations mais leur provenance se distingue par le plus fort taux d'étudiants ayant suivi la spécialité mathématiques en terminale (100% si l'on exclut les étudiants étrangers). Une analyse précise reste à mener pour interpréter les résultats par notion, qui sont bons en moyenne sauf pour la boucle `for` déjà citée.

Pour les universités ayant permis le passage du test à distance, les interprétations doivent rester prudentes et tenir compte du taux de réponse (voir figure 1) qui, s'il est faible, introduit un biais de sélection. L'analyse des différences de score entre établissements doit aussi tenir compte des spécificités des parcours qui varient de la L1 informatique au portail généraliste.

La licence informatique de l'université de Montpellier a passé le test en distanciel avec un taux de répondants un peu inférieur à la moitié (45%). Le score très bon (76,5%), qui doit être interprété avec une très grande prudence, peut en partie s'expliquer par le taux élevé d'étudiants ayant suivi les spécialités NSI

et maths au moins en première (57% en première pour NSI et 84% en mathématiques) parmi les répondants. Une très grande majorité des répondants ont trouvé que le niveau du test se situait entre très facile et normal (plus de 80%). L'étude montre que les notions vues en première et en terminale semblent majoritairement acquises. Comme pour les autres universités, ce sont les boucles `for` qui semblent poser le plus de difficultés contrairement aux boucles `while`.

L'université Grenoble Alpes (UGA) est l'établissement qui a fait passer le test au plus grand nombre d'étudiants (273 sur 368 primo-entrants + 204 redoublants, soit un taux de réponse global de 47,7%) réunis au sein de 5 portails intégrant deux à quatre disciplines scientifiques ; 3 (dont 2 internationaux) sont sélectifs et à petits effectifs ; l'UE « algorithmique et programmation » est obligatoire dans ces 5 portails. Le recueil de données n'ayant pas collecté le choix de portail, l'analyse ne peut pas les distinguer. C'est l'un des axes d'amélioration du questionnaire pour 2026 de tenir compte du portail d'inscription.

## Conclusions et perspectives

Cette première expérimentation du test PIKA dans les universités françaises donne déjà un résultat significatif par rapport au suivi de la spécialité NSI au lycée. L'obtention de résultats plus globaux dépendra de la généralisation du test à plus de formations, incluant les Bachelor Universitaire de Technologie (BUT), et d'établissements pour la rentrée 2026. Une meilleure anticipation de la mise en œuvre devrait permettre à plus d'établissements de réaliser ce test en présentiel. Une condition exprimée pour que ce test soit adopté à plus grande échelle est qu'il permette de donner un feedback à chaque étudiant sur ses connaissances et compétences à l'issue du test. Cela n'a pas été possible cette année à cause de contraintes techniques de la plateforme utilisée. Pour l'an prochain, il est prévu que la plateforme de test monte en capacité et permette cette fonctionnalité. Le test sera aussi amélioré en diminuant le nombre de questions et en diversifiant les questions portant sur la boucle bornée. Une étude des corrélations entre les données publiques issues de Parcoursup [5] et celles issues du test PIKA devrait permettre de caractériser plus finement les publics accueillis par chaque établissement et d'analyser les résultats au test en fonction de ces données.

La volonté que le test reste anonyme est fondée sur son objectif de mesurer globalement un niveau moyen de connaissances et de compétences pour adapter pour tous le premier cours de programmation à l'entrée à l'université. La question se pose cependant du possible usage du test pour le positionnement des étudiants, et la mise en place de parcours différenciés selon la provenance des étudiants, leur réussite à un test ou leur choix. C'est une perspective de cette étude préliminaire : la différence significative de résultats à un test de connaissances et de compétences en programmation à l'entrée à l'université en fonction de la spécialité suivie au lycée, ne nous laisse pas indifférents en tant que responsables de filières universitaires en informatique et nous incite à poursuivre cette réflexion collective vers l'évolution de l'accueil des étudiants primo-entrants souhaitant s'engager dans des études d'informatique à l'université.

## Références

1. Bolland, S., Haraldsrud, A., Jensen, S., Strömbäck, F., Styve, A., Tøssebro, E., Valseth, E., Strømme, T. : The nordic prior knowledge test in programming : Motivation, development and preliminary results. In : Norsk IKT-konferanse for forskning og utdanning. 4 (2024). <https://doi.org/10.5324/nikt.6216>
2. Bolland, S., Popov, A., Eide, T., Kordts, R., Strømme, T. : Fundamentals of norwegian cs1. In : Norsk IKT-konferanse for forskning og utdanning. 4 (2024). <https://doi.org/10.5324/nikt.6217>
3. Bolland, S., Strømme, T. : How introduction of programming to the k-12 curriculum influences prior knowledge for cs1. In : Proceedings of the ACM Global on Computing Education Conference 2025 Vol 1. p. 22–28. CompEd 2025, Association for Computing Machinery, New York, NY, USA (2025). <https://doi.org/10.1145/3736181.3747147>
4. Hoarau, S., Declercq, C., Chane-Lune, S. : A Proposal for a Programming Skills Framework Integrating Computational Thinking and Python Programming Concepts. In : Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education. vol. 1, pp. 417–423. ACM, Nijmegen, Netherlands (Jun 2025). <https://doi.org/10.1145/3724363.3729025>
5. Ministère de l'enseignement supérieur de la recherche et de l'espace : Parcoursup : propositions d'admission dans l'enseignement supérieur des élèves de terminale diplômés du baccalauréat général selon leurs enseignements de spécialité (2025), <https://data.enseignementsup-recherche.gouv.fr/explore/dataset/fr-esr-parcoursup-enseignements-de-specialite-bacheliers-generaux-2/>
6. Selby, C., Woollard, J. : Computational thinking : the developing definition (2013), <https://eprints.soton.ac.uk/356481/>
7. Wing, J.M. : Computational Thinking. Communications of the ACM **49**(3), 33–35 (2006). <https://doi.org/10.1145/1118178.1118215>

## Remerciements

Dans le cadre de ce travail, Martin Rosalie a bénéficié du soutien du Laboratoire d'Excellence (LABEX) TULIP (ANR-10-LABX-41) et de l'École Universitaire de Recherche (EUR) TULIP-GS (ANR-18-EURE-0019).