

# Réponse à la consultation sur le projet de programme ISN

Ce document constitue la réponse de la *Commission inter IREM Lycée* à la consultation sur le projet de programme de la spécialité ISN de terminale S.

## □ Les objectifs généraux

Les sciences du numérique peuvent et doivent contribuer, aux côtés des autres disciplines et en particulier des autres sciences, à la formation de l'esprit, à éduquer les élèves à la réflexion et au raisonnement. À ce titre elles méritent une place de choix au lycée comme discipline de formation intellectuelle généraliste. Par formation intellectuelle des lycéens nous entendons une mission plus ambitieuse que l'objectif nécessaire d'adapter les futurs citoyens et les futurs professionnels à la société d'aujourd'hui. Le préambule du programme de la spécialité ISN pourrait affirmer plus fortement cette ambition.

## □ Les modalités pédagogiques

La passivité de beaucoup d'élèves face aux apprentissages interroge fortement les enseignants et les rend attentifs à tout moyen pédagogique visant à rendre ces élèves plus actifs. La pédagogie de projet, mode d'enseignement privilégié par le programme, fait partie de ces moyens alternatifs. Une telle approche est séduisante et paraît adaptée à la nature de cet enseignement. Il nous semble toutefois qu'une telle approche ne s'improvise pas et nécessite une conception en amont très rigoureuse du dispositif sur l'ensemble de l'année. Sinon, le risque est grand de ne pas maîtriser les objectifs d'apprentissage. De nombreux écueils sont certainement à éviter pour conduire à bien une telle démarche jusqu'à son terme et ne pas éprouver un sentiment d'échec après l'investissement et l'enthousiasme initiaux. La formation des enseignants à ce type de pédagogie nous paraît être un élément clé de la réussite à moyen terme de cet enseignement de spécialité et donc de l'introduction de l'informatique au lycée. La conception pédagogique d'activités et de projets attractifs, formateurs et réalistes selon une progression pertinente est un travail important et difficile et qui devrait être envisagée de façon collective par des groupes d'enseignants en collaboration avec des formateurs. Nous suggérons l'idée de concevoir la formation initiale des enseignants elle-même sous la forme d'une pédagogie de projet, où les projets consisteraient à concevoir des activités et projets pour les élèves. Autrement dit une vision récurrente de la pédagogie de projets !

La mise en activité des élèves, la réalisation de petits projets et la pratique de la programmation ont pour but de susciter l'intérêt des élèves et de les familiariser dans le maniement des machines programmables, non pas sous leurs formes d'applications sophistiquées et contingentes mais dans leur aspect universel et fondamental. Il faut alors profiter de cet intérêt et de cette pratique des élèves pour, à chaque occasion qui se présente, dégager des idées importantes, faire émerger des concepts, mettre en évidence les principes des sciences du numérique. La pédagogie de projet ne devrait pas conduire à une interprétation a minima du programme avec une succession de séances de programmation systématiques et peu formatrices. Pour les élèves qui n'aborderont plus dans leurs études les sciences du numérique, la pratique de la programmation n'est pas une fin en soi mais le moyen de comprendre le monde numérique dans ses aspects les plus fondamentaux.

## □ Les éléments du programme

À la première lecture, ce programme peut apparaître trop chargé en comparaison du volume horaire imparti à cet enseignement. Il faut bien sûr tenir compte des petits losanges qui identifient des approfondissements qui ne sont pas exigibles. L'idée de ces items facultatifs nous semble bonne et mérite d'être conservée car elle contribue à la souplesse de ce programme. Toutefois la présence de ces approfondissements contribue sensiblement à l'impression de surcharge. Une solution éventuelle pourrait être de rédiger ces approfondissements dans une taille de police un peu plus petite afin de mieux faire ressortir les notions fondamentales et les capacités exigibles.

À juste titre le programme affirme en caractères gras que les quatre notions fondamentales, bien que présentées séquentiellement, doivent être considérées globalement. C'est une affirmation capitale et nous pensons qu'il serait très utile dans ce paragraphe d'argumenter cette nécessité et de l'illustrer par des exemples. Par ailleurs une bonne compréhension de cet impératif fera aussi percevoir avec plus de justesse la faisabilité de ce programme dans l'horaire imparti. Par ailleurs, il n'est pas nécessaire de parler d'emblée de ces quatre notions aux élèves. Il s'agit plutôt de faire émerger progressivement ces concepts qui s'avèrent utiles pour structurer nos idées.

Sous les quelques réserves que nous indiquons ci-dessous, nous pensons qu'il est donc possible de concevoir à partir de ce programme un ensemble cohérent d'activités pratiques, en particulier de programmation, qui pourront servir de matière première aux enseignants pour amener leurs élèves à se forger une représentation pertinente des concepts fondamentaux de l'informatique.

### ▷ Représentation de l'information

Les signaux émis par un sémaphore, la position des pièces sur un échiquier, une figure constituée d'arcs de cercles et de segments, la forme d'un labyrinthe sont aussi des informations. Elles sont moins abstraites qu'un nombre ou un caractère et plus faciles à coder qu'une image ou un son. Qu'entend-on par forme du labyrinthe ? Le codage de ces informations est relié à l'usage que l'on veut en faire. Coder une information consiste à l'abstraire, à la modéliser, en négligeant éventuellement les aspects qui ne sont pas pertinents par rapport au problème qui nous concerne. Ce codage a une part d'arbitraire mais tous les codages ne se valent pas car le traitement de l'information peut être plus ou moins simple selon le codage retenu. Par ailleurs, les codes d'aujourd'hui portent la marque des choix antérieurs liés à des besoins historiques. L'évolution du code ASCII vers un code plus universel en est un exemple.

Il est intéressant de souligner la nature universelle du codage des nombres et des caractères : toute information peut d'abord être codée sous la forme de nombres ou de caractères qui eux mêmes sont codés ensuite sous forme binaire. Ceci permet de comprendre qu'une machine peut calculer sur toute sorte d'objets et justifie le fait de parler de sciences du *numérique* à propos de l'informatique, c'est-à-dire de la science du traitement de l'*information*. Le codage intermédiaire sous la forme d'un texte permet de rendre l'information lisible par un humain.

Les portes logiques constituent un cadre particulièrement propice pour mettre en évidence la façon dont des dispositifs physiques peuvent être conçus dans le but d'effectuer des calculs ; calculs qui n'ont de signification que pour les utilisateurs de ces dispositifs.

L'utilisation d'un logiciel de modélisation d'images peut donner lieu à des activités très intéressantes mais ne mérite peut-être pas de figurer comme capacité

exigible.

Compression : La capacité mentionnée pourrait être plus explicite. L'élève devrait être capable de *choisir la compression* d'une donnée adaptée à l'utilisation envisagée.

## ▷ Algorithmique

Dans le cadre d'une pédagogie de projets, les élèves vont se retrouver confrontés d'eux-mêmes à des problèmes nécessitant de leur part l'invention de solutions algorithmiques. C'est la tâche du professeur de concevoir et proposer des projets susceptibles de générer des problèmes algorithmiques intéressants. Idéalement, les différents algorithmes imaginés par les élèves peuvent alors être comparés et une solution experte proposée par le professeur peut éventuellement être étudiée. Ainsi, il ne nous semble pas opportun que le programme fixe de façon aussi précise les algorithmes qui doivent être connus des élèves. Les algorithmes classiques de recherche et de tri sont sans doute incontournables car ils sont omniprésents dans les logiciels et parce qu'ils se prêtent bien à des comparaisons. Mais nous ne voyons pas les raisons de privilégier le tri par sélection plus que le tri par insertion ou le tri à bulles qui peuvent tout aussi facilement venir à l'esprit des élèves. Nous pensons, au contraire, intéressant de placer les élèves en situation (tri d'un jeu de cartes par exemple) afin de leur permettre d'élaborer différentes solutions algorithmiques. En plus d'illustrer le caractère universel des algorithmes, cela peut aussi être l'occasion de sensibiliser les élèves à la notion de performance d'un algorithme, en se gardant bien de tout développement théorique.

La récursivité est une possibilité offerte par beaucoup de langages mais c'est aussi une idée, une façon particulière de penser et de résoudre un problème. C'est une belle idée qui peut séduire beaucoup d'élèves et qui mérite d'être reconnue comme un type d'algorithmes.

Les protocoles et les mécanismes de routage sont aussi des algorithmes et il serait peut-être préférable de les placer dans cette section en y ajoutant un troisième encadré : Algorithmes distribués. Le caractère multiforme et omniprésent de cette notion fondamentale serait alors mieux mis en évidence.

Nous pensons donc que cette section du programme devrait valoriser davantage la notion d'algorithme. Il ne s'agit pas bien entendu de faire des cours théoriques d'algorithmique mais de faire réfléchir les élèves à partir de leurs productions concrètes. Lorsque les élèves travaillent à leurs projets, écrivent et mettent au point des lignes de code, ils produisent des algorithmes sans en avoir véritablement conscience, à la façon dont Monsieur Jourdain écrit sa prose. Il nous semble important que le professeur leur montre du doigt ces algorithmes chaque fois que l'occasion s'en présente afin qu'ils deviennent pour eux des objets de réflexion sur lesquels ils puissent porter un regard critique : quels sont ses contours ? quels sont ses données et ses résultats ? est-il correct ? puis-je l'améliorer, le rendre plus efficace ? ai-je déjà produit ou rencontré un algorithme similaire ? est-il réutilisable dans d'autres situations ? est-il adaptable, généralisable ?

Les progrès de l'informatique ne sont pas uniquement dûs aux progrès du matériel et à sa loi de Moore, l'invention d'algorithmes efficaces joue également un rôle très important. Il serait bon que les élèves en prennent conscience et sachent que la recherche d'algorithmes performants est une activité essentielle de l'informatique. Ceci peut contribuer à l'intérêt des élèves pour cette discipline.

## ▷ Langages et programmation

Dès que les élèves ont acquis une familiarité suffisante avec le langage et l'environnement de programmation qu'ils utilisent en classe, il nous paraît indispensable, afin de combattre la pensée magique, que soit clairement posée la question : *Que se passe-t-il lorsque je clique sur le bouton « compiler » ?* Il s'agit ici d'établir une relation causale suffisamment convaincante pour les élèves entre le texte de leur programme d'une part et le comportement de la machine d'autre part. Il doit être possible, en éliminant les détails techniques non essentiels, de parler de langage de haut niveau, de fichier source, de compilateur, de fichier exécutable, de langage machine. Si le langage utilisé est interprété l'explication devra être adaptée.

On peut souligner qu'un programme est d'un certain point de vue une information, une donnée comme une autre lue par une machine. Le fait que cette machine puisse lire un programme fait d'elle une machine universelle.

Lorsque les élèves ont fait l'expérience de l'existence quasi inéluctable du bug, la question doit être posée à la classe : *Comment s'assurer que tous les bugs ont été éliminés ?* Il s'agit de les sensibiliser au problème de la correction d'un programme. Le test d'un programme sur quelques données ne suffit pas pour le mettre au point. La mise en parallèle entre la sensibilité du programme à la moindre erreur et le nombre considérable de lignes de code nécessaires pour réaliser un logiciel permet aux élèves de prendre conscience de la complexité qui est en jeu et de la nécessité de disposer de méthodes.

Dans l'encadré *Fonctions* l'expression « passage d'arguments » peut laisser penser qu'il s'agit d'aborder la différence entre le passage par valeur et le passage par référence. Cette notion délicate ne devrait pas apparaître dans le programme même si certains élèves pourront y être confrontés en pratique à travers leurs projets.

Dans l'encadré *Langages de description*, le programme souligne à juste titre le double usage du langage, mais il serait bien de faire remarquer aux élèves qu'il en est de même de tout langage de programmation et plus généralement de toute information numérique.

## ▷ Architectures matérielles

Cette partie nous paraît être la plus délicate à transmettre aux élèves. L'enjeu principal est de leur montrer, à partir de leur connaissance de la physique, comment il est possible de stocker de l'information sur un support matériel (CD-ROM, code barre, bande magnétique d'une carte bleue, . . .) et surtout comment un système physique peut lire cette information, effectuer des calculs et modifier l'information. Des exemples de machines mécaniques élémentaires pourraient montrer la possibilité même d'existence d'un tel système physique. L'observation d'une boîte à musique est un magnifique point de départ pour introduire la notion d'information numérique. L'étude des principes mécaniques de la Pascaline par des élèves seraient également instructive. Il serait utile pour cela que les enseignants puissent disposer de supports pédagogiques spécialement conçus pour eux. Il est également instructif de faire remarquer aux élèves que d'une certaine façon on commet un abus de langage lorsque l'on dit qu'une machine effectue des calculs et manipule de l'information : c'est la manière dont l'utilisateur interprète le comportement de la Pascaline qui détermine si elle a effectué une addition ou une soustraction.

Tout ce travail est difficile mais l'enjeu en vaut la peine : il s'agit de démystifier l'ordinateur en le réduisant à un banal système physique au comportement com-

plètement spécifié par l'homme et parfaitement déterministe. Si ce travail peut être mené alors la présentation descriptive des composants de base d'un ordinateur prendra beaucoup plus du sens pour les élèves. Plutôt que de s'appuyer sur une machine « réelle » il serait préférable de présenter une machine fictive dans laquelle les aspects techniques seraient gommés au profit des principes fondamentaux. Là encore, des supports pédagogiques adaptés sont nécessaires pour faire comprendre le plus simplement possible ces principes.

La mémoire et la puissance de calcul mondiales dans les années 1950 étaient bien inférieures à celles du plus petit des téléphones actuels!<sup>1</sup> La loi de Moore mérite d'être expliquée aux élèves et illustrée par des exemples marquants. Les conséquences de cette « loi » quantitative sont qualitatives. Elle a provoqué et provoquera encore des mutations sociétales majeures. Une réflexion sur ce sujet pourrait être menée en concertation avec les professeurs des disciplines des sciences humaines.

Un autre travail interdisciplinaire possible, en collaboration avec le professeur de philosophie, serait d'initier une réflexion sur les idées de complexité et d'émergence. Les élèves pourraient s'appuyer sur leurs nouvelles connaissances en informatique, mais aussi sur leurs connaissances en biologie pour étayer une réflexion sur les frontières entre les notions de machine, d'être vivant, de pensée, de conscience.

Les réseaux constituent aujourd'hui un domaine essentiel de l'informatique. Il nous semble toutefois que les notions qui y sont associées ne sont pas aussi fondamentales que celles qui sont abordées dans les autres parties du programme. L'élève intéressé qui aura bien assimilé ces autres parties disposera de bases pour comprendre les questions liées aux réseaux. Il nous paraît donc souhaitable de retirer la partie réseau du programme, ou du moins de la considérer comme un approfondissement, afin de dégager plus de temps pour approfondir les concepts fondamentaux. Cela n'interdit pas au professeur d'illustrer certaines notions fondamentales à partir des réseaux. L'étude des composants d'un ordinateur fait déjà apparaître des mécanismes de communication entre ces composants. Des mini-projets peuvent amener les élèves à réaliser des petits programmes qui communiquent entre eux et la notion de protocole peut être dégagée si l'occasion s'en présente.

---

<sup>1</sup>Revue TDC n°997 - Gérard Berry