

Terminale NSI Activité 3

Durée :	1h
Classe :	Entière
Objectifs :	Vérifier que les notions de fonction itérative et récursive sont connues

Description

Le but de cette activité est de vous faire passer une évaluation formative : la note que vous obtenez est donnée à titre indicatif. L'important est vraiment que vous puissiez vous situer par rapport aux compétences que l'on attend de vous en spécialité NSI en classe de Terminale !

Exercice

Voici 5 questions concernant la récursivité. Les compétences évaluées sont les suivantes :

- Question 6 : maîtriser les différences entre la programmation itérative (boucles bornées et non-bornées précisément) et la programmation récursive
- Question 7 : question de cours sur la récursivité (cadeau !)
- Question 8 : assurer la terminaison d'une fonction récursive (au niveau des paramètres)
- Questions 9 & 10 : assurer la terminaison d'une fonction récursive (au niveau des conditions d'arrêt)

Vous devez illustrer, par écrit, pour toutes les questions sauf la 7, votre réponse par un exemple (ou un contre-exemple, c'est selon). Vous avez le droit d'utiliser votre ordinateur mais à la fin vous devez rendre votre travail sur une feuille de papier. Ne vous cassez pas la tête pour les exemples : ils doivent être le plus naïfs possible ☺

Vrai Faux

6. La récursivité peut s'utiliser pour remplacer une boucle <code>while</code> mais ne peut pas remplacer une boucle <code>for</code> .	<input type="checkbox"/>	<input type="checkbox"/>
7. On peut parler d'un programme récursif par opposition à un programme itératif.	<input type="checkbox"/>	<input type="checkbox"/>
8. Les valeurs des paramètres d'une fonction récursive doivent être identiques à chaque appel pour assurer la terminaison.	<input type="checkbox"/>	<input type="checkbox"/>
9. Dans une fonction récursive, il peut y avoir plusieurs conditions d'arrêt.	<input type="checkbox"/>	<input type="checkbox"/>
10. Les valeurs passées en paramètres pour les appels récursifs doivent constituer une suite décroissante.	<input type="checkbox"/>	<input type="checkbox"/>

Rappels intuitifs (par analogie fumeuse avec la logique formelle) pour vous aider à répondre aux questions

- Si la proposition vous dit : « tous les A sont des B » et que vous pensez que c'est faux, il vous suffit de montrer qu'il existe au moins un A qui n'est pas un B, et vous avez gagné
- Si la proposition vous dit : « parfois, les A peuvent être des B », et que vous pensez que c'est vrai, il vous suffit de montrer qu'il existe au moins un A qui est aussi un B, et vous avez gagné

Rappel très important :

Il n'existe pas de moyen automatique pour savoir si un programme termine ou pas : le problème de l'arrêt est indécidable ! En d'autres termes, aucun programme ne peut déterminer **à coup sûr** si un programme qu'on lui passe en paramètre s'arrête ou pas (cf. *Alan Turing*). Heureusement, il nous reste toujours la solution de « faire tourner l'algorithme à la main » ce qui peut nous donner des indications sur son arrêt ou non (cf. raisonnement par récurrence).