

Atelier algorithme et langage naturel

Que font ces algorithmes ?

<p>Saisir $f(x)$ Saisir a m prend la valeur $f(a)$ p prend la valeur $f(a)-am$ Afficher m Afficher p</p>	<p>Entrées Saisir n : entier naturel supérieur à 2 Saisir une liste de réels : x_1, x_2, \dots, x_n Traitement Pour i de 1 jusqu'à $n-1$ j prend la valeur n Tant que $i < j$ Si $x_j < x_{j-1}$ alors échanger x_j et x_{j-1} dans la liste FinSi j prend la valeur $j-1$ FinTantQue FinPour Sortie Afficher la liste x_1, x_2, \dots, x_n</p>	<p>$0 \rightarrow A$ $1 \rightarrow B$ $1 \rightarrow K$ TantQue $K \leq 5$ $A + B \rightarrow U$ $B \rightarrow A$ $U \rightarrow B$ $K + 1 \rightarrow K$ FinTantQue Afficher U</p>
---	---	---

Progression possible au lycée en algorithmique (compléter le tableau avec les classes, 2^{nde}, 1^{ère}, term.)

	Instructions	Comprendre	Modifier	Créer
Structure d'un algorithme et instruction	Entrée			
	Calcul ou traitement			
	Affectation			
	Sortie			
	Condition			
	Boucle « tant que »			
	Boucle « for »			
	Boucle « répéter »			

Donner des exemples d'algorithme en terminale dans les différentes parties du programme.

Des exemples dans vos classes cette année...

Les différentes questions possibles sur les algorithmes :

- Décrire un algorithme en langage naturel ;
- Comprendre et analyser un algorithme déjà écrit ;
- Valider et modifier un algorithme ;
- Trouver une erreur dans un algorithme ;
- Réaliser un algorithme simple.

Comment évaluer un algorithme ?

Voici des énoncés et des copies d'élèves à évaluer...

Soient u et v les suites définies par $u_n = 300 \times 1,5^n$ et $v_n = 310 + 20n$ et pour tout $n \in \mathbb{N}$.

Écrire un algorithme pour déterminer à partir de quel rang n , on a $u_n > v_n$.

On ne demande pas de trouver cette valeur.

$u_n = 300 \times 1,5^n$ $v_n = 310 + 20n$

Variables : u, v, n → nombres

DEBUT

n prend la valeur 0

u prend la valeur $300 \times 1,5^n$

v prend la valeur $310 + 20n$

 [Tant que $v > u$

n prend la valeur $n + 1$.

 Fin tant que.

 Afficher n .

FIN

Oliver Heaviside était un ingénieur anglais travaillant dans la télégraphie. En mathématiques, il a laissé une fonction H_1 aussi simple qu'utile, souvent appelée la « fonction marche ».

1 On donne ci-contre la représentation graphique de la fonction H_1 .

a. Déterminer $H_1(-2)$; $H_1(-0,5)$; $H_1(0)$ et $H_1(1,5)$.

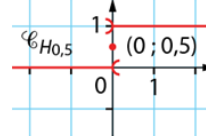
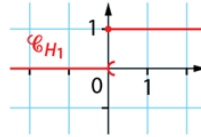
b. Écrire un algorithme nommé *Heavi* qui calcule l'image d'un réel quelconque par H_1 .

2 Pour plus de « symétrie » on utilise souvent une version modifiée de H_1 que l'on note $H_{0,5}$ et dont la représentation graphique est donnée ci-contre.

a. Quelle est la différence entre H_1 et $H_{0,5}$?

b. Modifier l'algorithme précédent pour obtenir un algorithme nommé *Heavi05* qui calcule l'image d'un réel quelconque par $H_{0,5}$.

c. Si l'on conserve des notations similaires, décrire la fonction notée H_0 , puis écrire l'algorithme nommé *Heavi0*.



Copie 1

1. a. $H_1(-2) = 0$
 $H_1(-0,5) = 0$
 $H_1(0) = 1$
 $H_1(1,5) = 1$

b. Entrer x .
 SI $x \geq 0$
 Alors $y = 1$
 Sinon $y = 0$
 Fin.
 Afficher y

2. a. La différence entre H_1 et $H_{0,5}$ est $f(0)$.

b. Entrer x .
 Si $x > 0$
 Alors $y = 1$
 Fin Si
 Si $x < 0$
 Alors $y = 0$
 Fin Si
 Sinon $y = 0,5$.
 Fin

c. La fonction H_0 , $f(0) = 0$.
 Pour
 Entrer x
 Si $x > 0$
 Alors $y = 1$
 Sinon $y = 0$
 Fin.

copie2

1) a. $H_1(-2) = 0$
 $H_1(-0,5) = 0$
 $H_1(0) = 1$
 $H_1(1,5) = 1$

b. Algorithme Heavi:

Début:
 affecter x
 Si $x \leq 0$
 alors $x = 0$
 Si $x > 0$
 alors $x = 1$
 Fin Si
 affecter x

2) a. Entre H_1 et $H_{0,5}$ la différence est que $H_{0,5}$ ne prend pas le 0 avec la paranthèse ouverte sur le graphique. Il y a un trou en 0.
 RAS

b. Algorithme Heavi05

Début
 affecter x
 Si $x < 0$
 alors $x = 0$
 Si $x > 0$
 alors $x = 1$
 Fin Si
 affecter x

Exemple de grille pour l'évaluation en algorithmique

Critère	Excellent	Bon	Moyen	Insuffisant
Respect des bons usages But visé par l'algorithme est explicité, des commentaires précisent le déroulement. Les variables ont des noms bien choisis.	Aucune erreur	De petits détails sont négligés. Le but est difficile à déterminer	Des détails manquent, mais le programme tente quand même d'accomplir Ses fonctions essentielles	Ne répond pas au problème posé. Objectif impossible à déterminer
Correction du code L'algorithme fonctionne	Fonctionne correctement dans tous les cas	Fonctionne pour des données (entrées) standard mais échecs mineurs sur des cas particuliers	Échoue pour des données (entrées) standard, mais pour une raison mineure	Échoue pour des données (entrées) standard, pour une raison importante
Interface utilisateur (entrées, sorties) Claire et commode	Aucune faute	1-3 fautes mineures	Plus de trois fautes mineures ou une faute majeure	Plus d'une faute majeure