Salut à toi,

Tu trouveras dans cette fiche les points clés de l'article « Comprends ça pour savoir faire un algorithme ! » que tu peux retrouver sur le site à cette adresse : www.lesmathsentongs.com/savoir-faire-un-algorithmesavoir-faire-un-algorithme

Si tu souhaites aller plus loin ou qu'il y a toujours des trucs pas clairs pour toi, viens me poser tes questions dans les commentaires de l'article ou sur les réseaux !

Au plaisir de t'aider à Réussir,

Steven.

Les Points Clés de l'article

- **Pour savoir faire un algorithme, il faut déjà savoir répondre à 2 questions :** Qu'est-ce qu'une variable ? Quels types d'instructions utiliser pour ordonner les opérations ?
- **Une variable est un ‹‹ boite ›› dans laquelle stocker de l'information.** Cette information peut prendre différentes formes : entiers, réels, caractères, mots...
- **Tu dois connaître les 3 instructions suivantes :** la condition Si/Alors/Sinon, la boucle Pour et la boucle Tant que. Avec ça tu pourras faire tous les algos niveau lycée.
- A la fin de l'article, tu trouveras l'exemple d'un algorithme qui permet de calculer la somme partielle d'une suite Un définie différemment suivant si n est pair ou impair.

Les Variables et leurs types

Ce que tu dois savoir :

- Une variable est un **élément** de ton algorithme dans lequel tu vas **stocker de l'info**,
- Comme son nom l'indique, **la valeur stockée peut varier** tout au long de l'algo
- Les 3 types que tu dois savoir utiliser sont **les entiers, les réels et les complexes.**

Les 3 Instructions à bien comprendre

1) la Condition « Si/Alors/Sinon »

- Permet de définir **une opération dans un cas** et **une autre dans le cas contraire**,
- Ex : Au moment de traverser la route.
  Si il n'y a pas de voiture qui arrive **Alors** je traverse **Sinon** j'attends.

2) la Boucle « Pour »

- Permet de **répéter une opération X fois**, avec X connu et fixé,
- A chaque itération, **l'opération doit être la même mais ses paramètres peuvent changer,**
- Par ex : Si l'opération est une addition, ça restera une addition pour chaque itération, mais le chiffre ajouté pourra changer.

3) la Boucle « Tant que »

- Permet de **répéter une opération jusqu'à ce qu'une condition soit vérifiée**,
- Par ex : **Tant que** tu n'as pas 16 de moyenne en Maths, tu révises 2h par jour.

Dans l'article je t'ai détaillé un exemple d'algorithme pour calculer la somme partielle d'une suite.

D'autres exemples d'algorithmes !

⇨ Somme d'une suite géométrique

- Ecris l'algorithme qui te permet de **calculer la somme des N premiers termes** de la suite géométrique de raison **q=1/2** et de terme initial $\mathbf{u_0 = 2}$.
- Vérifie alors que le résultat est bien **le même que** celui donné par **la formule du cours**

⇨ Estimation de la limite d'une fonction en +∞

- Ecris un algorithme qui te permettrait **d'estimer la limite d'une fonction f(x) en +∞**
- Vérifie sur des exemples dont tu connais la limite que ton algo fonctionne bien.

⇨ Tracer un dodécagone régulier

- Ecris un algorithme qui permettrait de tracer un dodécagone régulier dont un des sommets est sur l'axe des abscisses.
- Programme-le dans GéoTortue par ex. pour le tester ! http://geotortue.free.fr/

Essaie de faire ces algos toi-même ! Dans la page suivante, tu trouveras les solutions.

Au plaisir de t'aider à Réussir,

Steven

EX 1 :

Rien de spécial à dire pour cet exemple

Entrées : u et s réels, n et N entiers
Initialisation : Affecter à u et s la valeur $u_0 = 2$
Traitement :
Lire la valeur de N;
pour *n allant de 1 à N* **faire**
| Affecter à s la valeur $s + u \times (1/2)$
fin
Sorties : Afficher la valeur de la somme s et la différence à la valeur théorique $2 \times \dfrac{1 - q^{N+1}}{1 - q}$.

EX 2 :

Tout est hyper simplifié pour cet exemple.

Si la fonction est périodique de période 1, cet algo ne fonctionne pas. Pourquoi ? Comment éviter ce problème ?

Entrées : $x, f0, f1$ et ϵ réels
Initialisation : ;
Affecter à x la valeur 0;
Initialiser $f0$ à la valeur $f(x)$;
Initialiser $f1$ à la valeur $f0 + 1$;
Traitement :
Lire la précision voulue de l'estimation ϵ;
tant que *$(|f0 - f1| < \epsilon)$* **faire**
| Affecter à $f0$ la valeur $f(x)$;
| Affecter à x la valeur $x + 1$;
| Affecter à $f1$ la valeur $f(x)$;
fin
Sorties : Afficher l'estimation de la limite $f0$.

EX 3 :

Rien de fou pour cet algorithme mais tu vois que ça peut aussi s'appliquer à de la géométrie !

Entrées : x, y réels
Initialisation : Se placer en $(0, 1)$;
Traitement :
pour *(n allant de 1 à 12)* **faire**
| Affecter à x la valeur $\cos(n \times 360/12)$;
| Affecter à y la valeur $\sin(n \times 360/12)$;
| Aller au point (x, y) en traçant le chemin;
fin

Viens me poser tes questions et me faire tes remarques !

Like ma page Facebook

Suis-moi sur Twitter

Abonne-toi à la chaîne YouTube

Laisse un commentaire sur le site

Et aide-moi à aider tes amis en partageant avec eux !

les Maths en Tongs

Le bon état d'esprit pour Réussir !