

Notion de base en algorithmique

[Concepts importants en informatique](#)

[Pourquoi un cours "d'algo" ?](#)

[Les problèmes fondamentaux en algorithmie](#)

[Exemples d'écriture algorithmique](#)

[Calcul de la moyenne d'un tableau de notes](#)

[Calcul du carré d'un nombre que l'utilisateur
fourni](#)

[Les étapes d'un algorithme](#)

[Conseils d'écriture du langage algorithmique](#)

[La phase d'analyse](#)

[Exemple d'énoncé d'un problème](#)

[Résolution algorithmique de l'énoncé](#)

Concepts importants en informatique

Le mot algorithme vient du nom latinisé du mathématicien perse AlKhawarizmi, écrivant en langue arabe.

Un algorithme prend des données en entrée, exprime un traitement particulier et fournit des données en sortie.

Tout traitement demandé à la machine, par l'utilisateur, est effectué par l'exécution séquencée d'opérations. Il y a 4 ingrédients de base, qui sont les instructions, boucles, variables et les conditions.

Le tout est appelée un programme ou un logiciel.

Pourquoi un cours "d'algo" ?

Pour obtenir de la «machine» qu'elle effectue un travail à notre place.

Problème : expliquer à la «machine» comment elle doit s'y prendre.

Besoins :

- savoir expliciter son raisonnement,
- savoir formaliser son raisonnement,
- concevoir (et écrire) des séquences d'instructions qui décrivent comment résoudre un problème particulier, autrement dit : des algorithmes.

L'écriture algorithmique est un travail de programmation à visée universelle :

- un algorithme ne dépend pas du langage dans lequel il est implanté,
- ni de la machine qui exécutera le programme correspondant.

Exemples d'algorithmes de la vie courante :

- une recette de cuisine,
- une notice de montage de meuble en kit,
- la construction d'une voiture et son assemblage
- ...

Les problèmes fondamentaux en algorithmie

La complexité :

- En combien de temps un algorithme va-t'il atteindre le résultat escompté ?
- De quel espace a-t'il besoin ?

La complétude :

- Existe-t'il des tâches pour lesquelles il n'existe aucun algorithme ?
- Étant donnée une tâche, peut-on dire s'il existe un algorithme qui la résolve

?

La correction :

- Peut-on être sûr qu'un algorithme réponde au problème pour lequel il a été conçu ?

Exemples d'écriture algorithmique

Calcul de la moyenne d'un tableau de notes

- Entrées : un entier n , un tableau de n nombres à virgules
- Sortie : la moyenne des nombres du tableau

variables *nombreTemporaire* est un entier ; *monNombre* est un nombre à virgule ;
début

initialiser *monNombre* avec 0 pour *nombreTemporaire* de 1 à n , faire :
 ajouter le *nombreTemporaire* ème élément du tableau à *monNombre*
fin de pour

retourner le résultat de l'opération mathématique *monNombre* diviser par n
fin

Calcul du carré d'un nombre que l'utilisateur fourni

- Entrées : un entier *unNombre*, un entier *sonCarre*
- Sorties : proposer une sortie écran de la phrase "Le carré de *unNombre* est *sonCarre*"

variables *unNombre*, *sonCarre* : entiers ; {déclarations : réservation d'espace mémoire}
début

 {préparation du traitement}

afficher("Quel nombre voulez-vous élever au carré ?")

saisir(*unNombre*)

$\text{sonCarre} \leftarrow \text{unNombre} \times \text{unNombre}$ {traitement : calcul du carré}

afficher("Le carré de", *unNombre*, "est", *sonCarre*) {présentation du résultat}

fin

Les étapes d'un algorithme

1. Préparation du traitement :
exprimer les données nécessaires à la résolution du problème
2. Traitement :
résolution pas à pas (décomposition en sous-problèmes si nécessaire)
3. Édition des résultats :
impression à l'écran, dans un fichier, ...

Conseils d'écriture du langage algorithmique

1. Avoir une écriture rigoureuse
2. Avoir une écriture soignée et structurée : respecter l'indentation des lignes et la convention de nomage
3. Commenter les algorithmes
4. Rechercher l'efficacité de ce que l'on écrit : il existe plusieurs solutions algorithmiques à un problème posé

La phase d'analyse

Elle consiste à extraire de l'énoncé du problème des éléments de modélisation.

La technique est donc de distinguer les différentes étapes (déclaration des données, lecture/écriture des données, traitement à réaliser). Pour commencer, il est conseillé soit de noter séparément ces éléments, soit de vous munir de surligneurs de 3 couleurs différentes afin de marquer visuellement ces éléments dans l'énoncé (on a tendance à associer le bleu pour le traitement à réaliser, le rouge pour les données en entrée du problème et le vert pour les résultats en sortie).

Exemple d'énoncé d'un problème

“On souhaite *calculer et afficher, à partir d'un prix hors taxe saisi, la TVA ainsi que le prix TTC.*

Le montant TTC dépend du *prix HT* et du *taux de TVA (20%)*”

Résolution algorithmique de l'énoncé

Algorithme CalculerTVA {saisi un prix HT et affiche le prix TTC correspondant}

{déclarations : réservation d'espace mémoire}

constantes TVA : entier \leftarrow 20

Titre : chaîne de caractères \leftarrow “Résultat”

variables prixHT : nombre à virgule

prixTTC, montantTVA : nombre à virgule

début

{préparation du traitement}

afficher(“Prix HT :”)

saisir(prixHT)

$\text{prixTTC} \leftarrow \text{prixHT} * (1 + \text{TVA}/100)$ {traitement : calcul du prix TTC}

$\text{montantTVA} \leftarrow \text{prixTTC} - \text{prixHT}$ {traitement : calcul du montant de TVA}

afficher(Titre) {présentation du résultat}

afficher(prixHT, “ euros HT + TVA “ , TVA, “ devient “ , prixTTC, “ euros TTC”)

{présentation du résultat}

fin

Remarque : ce code est peu efficace et doit être optimisé.