

Algorithme

Définition

Un algorithme est une suite ordonnée d'instructions qui indique la démarche à suivre pour résoudre une série de problèmes équivalents.

Un algorithme doit obtenir une solution en un nombre fini d'étapes.

Exemples

Algorithmes de la vie courante

Exemple 1 : Une partie du mode d'emploi du téléphone Galaxy s4 mini

Comment effacer un nom dans mon répertoire

1. Depuis l'écran d'accueil, appuyez sur l'icône CONTACTS
2. Vous accédez au répertoire. Appuyez longuement sur le contact que vous souhaitez supprimer.

Au besoin lancez une recherche.

3. Un menu s'affiche. Appuyez sur Supprimer
4. Un message de confirmation s'affiche. Appuyez sur OK
5. Votre contact est supprimé. Appuyez sur la touche ACCUEIL
6. Vous revenez à l'écran d'accueil.

Lancez une recherche dans le répertoire.

1. Depuis l'écran d'accueil, appuyez sur l'icône CONTACTS
2. Vous accédez au répertoire. Appuyez dans le champ RECHERCHER.
3. À l'aide du clavier virtuel qui s'affiche, saisissez les premières lettres du nom du contact recherché.

Les contacts du répertoire sont filtrés en fonction des lettres entrées.

4. Appuyez sur le contact recherché.
5. Les détails de votre contact s'affichent.

Nous avons ici un algorithme qui indique la marche à suivre pour effacer un contact dans un téléphone. Cet algorithme fait appel si besoin est, à un autre algorithme permettant de rechercher un contact.

Cet algorithme est composé d'une suite ordonnée d'instructions (appuyez ..., appuyez longuement, faites une recherche...) qui manipulent des données (touche, icône, ligne de menu) pour réaliser la tâche désirée : supprimer un contact.

Exemple 2 : Trouver son chemin

(Exemple tiré du cours « Initiation à l'algorithmique » Jacques Tisseau)

Un touriste égaré interroge un passant

- Pourriez-vous m'indiquer le chemin de la gare, s'il vous plait ?
- Oui bien sûr : vous allez tout droit jusqu'au prochain carrefour, vous prenez à gauche au carrefour et ensuite la troisième à droite, et vous verrez la gare devant vous.
- Merci

Dans ce dialogue, la réponse du passant est une suite ordonnée d'instructions (allez tout droit, prenez à gauche, prenez la troisième à droite) qui manipulent des données (carrefour, rue) pour réaliser la tâche désirée (aller à la gare).

Quand on définit un algorithme, celui-ci ne doit contenir que des instructions compréhensibles par celui qui devra l'exécuter.

Autres définitions concernant les algorithmes

(Définitions tirées du cours « Initiation à l'algorithmique » Jacques Tisseau)

Validité d'un algorithme

La validité d'un algorithme est son aptitude à réaliser exactement la tâche pour laquelle il a été conçu.

Si l'on reprend le deuxième exemple : algorithme de recherche du chemin de la gare, l'étude de sa validité consiste à s'assurer qu'on arrive effectivement à la gare en exécutant scrupuleusement les instructions dans l'ordre annoncé.

Robustesse d'un algorithme

La robustesse d'un algorithme est son aptitude à se protéger de conditions anormales d'utilisation.

Pour notre exemple 2, qu'arrive-t-il si l'algorithme a été pensé pour un piéton alors que le « touriste égaré » est en voiture et que la « troisième à droite » est en sens interdit ?

Réutilisabilité d'un algorithme

La réutilisabilité d'un algorithme est son aptitude à être réutilisé pour résoudre des tâches équivalentes à celle pour laquelle il a été conçu.

Puis-je réutiliser mon algorithme de recherche du chemin de la gare, pour aller à la mairie ? Oui, si la mairie est à côté de la gare, sinon non.

Complexité d'un algorithme

La complexité d'un algorithme est le nombre d'instructions élémentaires à exécuter pour réaliser la tâche pour laquelle il a été conçu. Elle permet de prévoir le temps d'exécution d'un algorithme.

Si le « touriste égaré » est un piéton, la complexité de l'algorithme de recherche de chemin peut se compter en nombre de pas pour arriver à la gare.

Efficacité d'un algorithme

L'efficacité d'un algorithme est son aptitude à utiliser de manière optimale les ressources du matériel qui l'exécute.

N'existerait-il pas un raccourci piétonnier pour arriver plus vite à la gare ?

Algorithmique

L'algorithmique permet ainsi de passer d'un problème à résoudre à un algorithme qui décrit la démarche de résolution du problème.

La programmation a alors pour rôle de traduire cet algorithme dans un langage « compréhensible » par l'ordinateur afin qu'il puisse exécuter l'algorithme automatiquement.

Construire un algorithme

Un algorithme est en général décomposable en trois parties :

- Un pré-traitement : entrée des données au clavier, initialisation des valeurs, . . .
- Un traitement : calculs, manipulation des données, . . .
- Un post-traitement : affichage des résultats sur l'écran, écriture dans un fichier, . . .

Un algorithme manipule des données, réalise des calculs avec ces données.

Comment un ordinateur fait-il pour traiter les données ?

L'ordinateur doit stocker les données qu'il manipule dans des zones de sa mémoire et il doit être capable de les retrouver. Il repère donc chaque zone de mémoire par une adresse associée à un nom.

Lorsqu'on utilise une donnée dans un algorithme, il faut donc penser que cette donnée, sera mémorisée dans une variable lorsqu'on traduira l'algorithme en un programme.

Une variable en programmation, est un espace de la mémoire de l'ordinateur:

- qui est désigné par un nom, que l'on choisit. Ce nom est l'identificateur de la variable.
- qui peut contenir une valeur (nombre, lettre, mot, tableau de nombres etc.)

Affectation : une affectation est l'attribution d'une valeur à une variable.

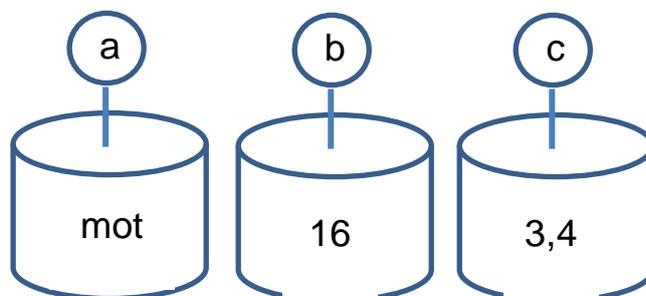
L'ordinateur mémorise la valeur à l'emplacement désigné par le nom de la variable.

Sur le schéma ci-dessous nous avons trois variables.

La variable de nom a contient une chaîne de caractère « mot ».

La variable de nom b contient une valeur numérique entière 16.

La variable de nom c contient une valeur numérique réelle 3,4.



Exercices à faire avec Algobox

Ces exercices permettent de prendre en main Algobox.

Pour chacun d'eux une petite vidéo permet de voir comment utiliser Algobox

Exercice 1 : écrire l'algorithme permettant de calculer la moyenne de trois nombres.

Exercice 2 : écrire l'algorithme permettant de calculer x^2 , x^3 et \sqrt{x}

Pour la sixième on peut choisir d'autres calculs : $2*a$, $2+a$ etc.

Exercice 3 : écrire l'algorithme permettant de calculer $(x + y)^2$ pour deux nombres donnés x et y .

Pour la sixième on peut choisir : $a + b$, $2 + a + b$ etc.

Exercice 4 : écrire l'algorithme qui affiche « Bonjour untel », untel étant le prénom de l'utilisateur de l'algorithme.

Traduire ces algorithmes en Scratch

Ces exercices permettent de prendre en main Scratch.

Pour chacun d'eux une petite vidéo permet de voir comment mettre en œuvre Scratch pour traduire l'algorithme.

Exercices d'entraînement à réaliser avec Algobox

(L'idée de base de ces exercices est tirée d'exercices de [France-IOI](#))

Exercices 1

C'est bientôt Noël et la mairie va faire passer le « Père Noël » dans l'école « Du sapin géant ». Celui-ci doit distribuer à chaque enfant, 3 petits paquets de friandises.

Pour cela, la mairie doit calculer le nombre de paquets de friandises à acheter.

Ce que doit faire votre algorithme :

L'école est formée de 4 classes, constituées respectivement de 25, 30, 27 et 22 élèves. Cependant, 8 élèves sont absents le jour de la distribution. Sachant que chaque élève présent doit recevoir 3 paquets, écrivez un algorithme qui calcule puis affiche le nombre total de paquets nécessaires, le nombre de paquets distribués et le nombre de paquets restants.

L'algorithme doit afficher :

Il faut acheter : xxxx paquets de friandises

Le jour de la distribution, il y a eu yyyyy paquets distribués

Il reste zzzz paquets.

Exercice 2

À la fin d'une course d'endurance, les élèves d'une classe sont épuisés.

Ils s'entraînent dur pour la course d'endurance qui aura lieu avant les grandes vacances.

Cette course durera trois jours. Chaque jour les coureurs devront enchaîner trois épreuves : 2 km de natation, 22 km de vélo et 5 km de course à pied.

Ce que doit faire votre algorithme :

Votre algorithme doit afficher la distance qu'aura parcourue un coureur à la fin du premier jour, à la fin du deuxième jour et à la fin de la course au bout des trois jours.

Exercice 3

La cour de récréation de l'école a bien besoin d'être refaite : d'une part la clôture actuelle est trop rouillée, et d'autre part il faudrait couvrir le sol en terre par de belles pierres pour éviter d'avoir de la boue dès qu'il pleut.

Le professeur de la classe fait participer ses élèves : il leur demande de calculer l'aire et le périmètre du carré que forme leur cour de récréation.

Les élèves n'ayant pas de règle assez longue, ils ont utilisé 4 bâtons de longueurs différentes pour mesurer le côté du carré. Afin d'aider les élèves dans leurs calculs, vous décidez d'écrire un algorithme.

Ce que doit faire votre algorithme :

La cour carrée a été mesurée avec quatre bâtons de longueurs respectives 17 m, 7 m, 5 m et 2 m. La longueur du côté de la cour est égale à 5 fois le premier bâton, plus 2 fois le second, plus 1 fois le troisième, plus 2 fois le quatrième.

Votre algorithme doit afficher deux lignes : la première doit contenir la surface de la cour, et la seconde ligne doit contenir son périmètre. Les résultats doivent être exprimés en mètres carrés et en mètres, respectivement, mais vous ne devez pas afficher l'unité après la valeur numérique.