**R1.01 : 03/09/2021**

Algorithme : Le mot vient d’un mathématicien perse 780.Al-Khwarizmi.

Un algorithme est la spécification d’un schéma de calcul, sous forme d’une suite finie d’opérations élémentaires obéissant à un enchainement déterminé.

Un algorithme est une procédure de calcul bien définie qui prend en entrée une valeur, ou un ensemble de valeurs, et qui donne en sortie une valeur, ou un ensemble de valeurs. C’est donc une séquence d’étapes de calcul qui transforment l’entrée en sortie.

Suite finies et non ambiguë et résoudre une classe de problèmes.

**Définition du cours : Un algorithme est la description d’une solution détaillant des opérations simples à enchainer pour résoudre un problème spécifique.**

**Problème : Définir le problème**

**Puis Description de la solution.**

**Exemple de la vie courante : Recette de cuisine ou explication d’un chemin à un tourisme**

**Exemple informatique : Un programme simple pour diviser deux nombres**

Il est important de cerner le contexte dans lequel l’algorithme s’exécutera.

Il existe deux types de langages pour décrire des algorithmes :

* Langage à pseudo code : Début, Fin, Ecrire, Si … alors, retour.
* Langage graphique : comme le pseudo code mais remplaces les mots par des symboles.

L’algorithmique c’est quoi ?

* Méthodes pour concevoir
* Techniques pour évaluer l’efficacité d’un algorithme
* Algorithmes « modèles » qui peuvent être utilisés pour résoudre rapidement et efficacement un problème.

Maitrise l’algorithmique c’est :

* Savoir concevoir les différentes briques composant un programme
* Savoir manier la structure logique d’un programme

**Analyse et spécification d’un problème algorithmique**

Pour analyser, identifier et spécifier les caractéristiques du problème à résoudre, une approche consiste à établir successivement ;

Spécifications externes : Visent à définir ce que le programmeur doit faire

Identifier les différents comportements possibles du programme :

Comportement idéal

Comportement Particulier

Comportement en cas d’erreur

Pour chaque comportement :

Ce que voit l’utilisateur

Les informations données du programme

Les résultats attendus par apport aux données entrées

**Les propriétés à chercher**

Qu’est-ce que l’informatique ?

Automatisation du traitement de l’information par l’ordinateur

Tous les programmes sont composés d’informations et de traitements

**Cours du 06/09/2021 :**

**Démarche de conception d’un algorithme :**

L’approche descendante consiste à décomposer un problème en sous problème qui, à leur tour, peuvent être re-décomposés autant de fois que nécessaire jusqu’à obtenir des sous problèmes faciles à résoudre.

L’algorithme consiste à résoudre un problème en décomposant ce problème en sous-problème.

La science informatique nous enseigne que tous problème pouvant être résolu par un algo.

Décomposition séquentielle

Décomposition alternative

Décomposition itérative

**Principe de la décomposition séquentielle :**

Consiste à résoudre successivement un ensemble de problème mais complexe.

Elle formalise :

* Quels sont les problèmes à résoudre
* L’ordre dans lequel il faut les résoudre

Problème : Comment se préparer le matin pour aller travailler

L’ordre des étapes est essentiel pour que la solution soit consistante et correct.  
Parfois l’ordre n’a pas d’importance  
Parfois l’ordre des étapes permet d’optimiser la solution

**Décomposition formalisme générale :**

Les problèmes et sous problèmes sont représentés par des rectangles ;

Les noms des problèmes et des sous problèmes sont renseignés à l’intérieur des rectangles

**Algorithmes doivent être visibles et compréhensibles**

**Prendre l’habitude de d’abord nommer les sous problèmes avant de décrire comment les résoudre**

La décomposition séquentielle est donc représenter par les doubles barres

Tout sous problèmes jugés trop complexe peut être à son tour décomposés en sous problèmes.

**Représenter les données et les résultats :**

Quel que soit le niveau de décomposition on mettra en évidence :

* Les données nécessaires
* Les résultats produits ou modifiés suite à la résolution du problème

D’un point de vue de la représentation algorithmique :

Les variables en données sont notés entre accolades à gauches

Les résultats sont notés entre accolades à droites

**Affectation :**

L’opération d’affectation est représenté algorithmiquement par une flèche : variable 🡨 expression

Expression peut être : Valeur, variable, fonction, un calcul

**Principe de la décomposition alternative**

Une décomposition alternative exprime le fait que la résolution d’un problème P de nature complexe équivaut à résoudre selon les cas le sous problème PA ou bien le sous problèmes P2 ou bien… le sous-problème Pn.

**Formalisme général :**

Voir cours

**Décomposition SI :**

Voir cours

**Décomposition SI – Sinon :**

Voir cours

**Décomposition SI –SI**

Voir cours

**Construction à proscrire :**

La construction ci-dessous (voir cours) est inconsistante : Pour une moyenne égale à 8, toutes les conditions spécifiés dans l’algorithme sont vraies

**Cas du switch :**

De nombreux problèmes ont des solutions dont la mise en œuvre dépend de la valeur d’une information

Exemple : Proposer un algorithme qui affiche en toutes lettre un chiffre dont la valeur est saisie au clavier. Comportement attendu 🡪 Entrer un chiffre 4, Vous avez saisi le chiffre quatre

Solution possible : Etudier la valeur du chiffre saisie et agir en conséquence

VOIR COURS

**NE FONCTIONNE QU’AVEC L’EGALITE // VOI SYNTAXE SUR COURS**