

COURS ALGORITHMIQUE ET PROGRAMMATION INFORMATIQUE

GRETA CFA PROVENCE – CICALA Lionel - 2020

RÉFÉRENCES

- Algorithmes D.E Knuth CSLI Publications 2011.
- Introduction a la science informatique G. Dowek Ed RPA 2010.
- Éléments pour une histoire de l'informatique, D.E Knuth CSLI Publications 2011.
- Cours et exercices corrigés d'algorithmique- J. Julliand Ed Vuibert Fev 2010.
- Algorithmique méthodes et modèles , P Lignelet Ed Masson 1988.

OBJECTIF DU COURS

- Notions de base en algorithmique.
- Types de données et lien avec la machine.
- Notion de sous-programmes et lien avec la compilation.
- Qualité :
 - nommage des variables, assertions, documentation ...
 - pré et post conditions.
- Structures algorithmiques fondamentales :
- Implantation des algorithmes dans un langage de programmation.
- Introduction au test unitaire, boîte noire.
- Algorithmes fondamentaux de recherche d'un élément, parcours, tri, ...
- Avoir une première notion des performances des algorithmes utilisés.

NOTION DE BASE EN ALGORITHMIQUE

CONCEPTS IMPORTANTS EN INFORMATIQUE

- **Algorithme** : mot dérivé du nom du mathématicien Al_Khwarizmi qui a vécu au 9^{ème} siècle, était membre d'une académie des sciences à Bagdad .
- Un algorithme prend des **données en entrée**, exprime un traitement particulier et fournit des **données en sortie**.
- **Programme** : série d'instructions pouvant s'exécuter en séquence, ou en parallèle (parallélisme matériel) qui réalise (**implémente**) un algorithme.

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 *explicit*er son raisonnement.
 - Savoir *formal*iser son raisonnement.
 - Concevoir (et écrire) des *algorithmes* :
 - Séquence d'instructions qui décrit comment résoudre un problème particulier.

ALGORITHME

- **Savoir expliquer** comment faire un travail sans la moindre ambiguïté.
- **Langage simple** : des instructions (pas élémentaires).
- Suite finie d'actions à entreprendre en respectant une chronologie imposée.
- L'écriture algorithmique : 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.

EXEMPLE D'ALGORITHMES

- Recette de cuisine

Recette de la pâte à crêpes

Il te faut :


250 g 50 cl 3 oeufs 1 pincée



1


Verse la farine et le sel dans le saladier.

2


Casse les oeufs, mélange avec la cuillère en bois et ajoute progressivement le lait sans cesser de tourner.

3


Laisse reposer la pâte pendant 1 heure.

4


Fais chauffer la poêle. Verse-y une louche de pâte. Répartie-la bien bougeant la poêle. Fais cuire une crêpe fine.

EXEMPLE D'ALGORITHMES

- Notice de montage de meuble en kit



EXEMPLE D'ALGORITHMES

- **Mathématiques** : problème $3n+1$: élémentaire mais redoutable
- Si n est pair, on le divise par 2 ;
- Si n est impair, on le multiplie par 3 et on ajoute 1.
- Est-il vrai que l'on finira tôt ou tard par tomber sur 1 ?

LES PROBLÈMES FONDAMENTAUX EN ALGORITHMIQUE

- **Complexité :**

- En combien de temps un algorithme va -t-il atteindre le résultat escompté?
- De quel espace a-t-il besoin?

- **Calculabilité :**

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

- **Correction :**

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

EXEMPLE DE LANGAGE ALGORITHMIQUE

Algorithme ElèveAuCarré

{Cet algorithme calcule le carré du nombre que lui fournit l'utilisateur}

variables unNombre, sonCarré: entiers

*{déclarations: réservation
d'espace-mémoire}*

début

{préparation du traitement}

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

saisir(unNombre)

{traitement : calcul du carré}

sonCarré \leftarrow unNombre \times unNombre

{présentation du résultat}

afficher("Le carré de ", unNombre)

afficher("c'est ", sonCarré)

fin

ETAPES D'UN ALGORITHME

- **Préparation du traitement :**
 - Données nécessaires à la résolution du problème.
- **Traitement :**
 - Résolution pas à pas.
 - Après décomposition en sous-problèmes si nécessaire.
- **Edition des résultats :**
 - Impression à l'écran.
 - Dans un fichier, etc.

LANGAGE ALGORITHMIQUE

- Il faut avoir une **écriture rigoureuse**.
- Il faut avoir une écriture soignée : respecter **l'indentation**.
- Il est nécessaire de **commenter** les algorithmes.
- **Il existe plusieurs solutions algorithmiques à un problème posé.**
- X Il faut rechercher **l'efficacité** de ce que l'on écrit.

Algorithme NomAlgorithme

{ ceci est un commentaire }

Début

... Actions

Fin

Algorithme Bonjour

{ il dit juste bonjour mais ...
en anglais ! }

Début

afficher('Hello world !!!')
ALaLigne

Fin

DÉCLARATION DES DONNÉES

- **Variable** : <nom de donnée> : **type**
- Instruction permettant de réserver de l'espace mémoire pour stocker des données
- Dépendant du type des données : entiers, réels, caractères, etc.)
- Exemples :
 - Variables** val, unNombre: **entiers**
 - nom, prénom : **chaînes de caractères**

DÉCLARATION DES DONNÉES

- **Constante**: <nom de donnée> : **type** ← valeur ou expression
- Instruction permettant de réserver de l'espace mémoire pour stocker une constante dont la valeur ne varie pas.
- Dépendant du type des données : entiers, réels, caractères, etc.)
- Exemples :
 - Constante** MAX : **entier** ← 10
 - DEUXFOISMAX : **entier** ← MAX x 2

LECTURE ÉCRITURE DE DONNÉES

- **Saisir** <nom de donnée, ...>
- **Afficher** <nom de donnée, ...>
- **Fonction** : Instructions permettant
 - • de placer en mémoire les informations fournies par l'utilisateur.
 - • De visualiser des données placées en mémoire
- **Exemples:**
 - Saisir (unNombre)
 - Afficher (« le nom est », nom, « et le prénom est », prénom)
 - Saisir (val)

PHASE D'ANALYSE

- Consiste à extraire de l'énoncé du problème (cahier des charges) les éléments de modélisation.
- **Technique** : Distinguer en soulignant de différentes couleurs quelles sont ces éléments :
 - Quel est le but du programme (**traitement à réaliser**).
 - **Données en entrée** du problème.
 - Où vont se situer 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 de :
 - Du prix HT
 - Du taux de TVA de 20,6 %

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 de :
 - **Du prix HT**
 - **Du taux de TVA de 20,6 %**
 - **Données en entrée**
 - **Données en sortie**
 - **Traitement à réaliser**

EXEMPLE ALGORITHME TVA

Algorithme CalculTVA

{Saisit un prix HT et affiche le prix TTC correspondant}

Constantes (TVA : réel) $\leftarrow 20.6$

(Titre : chaîne) \leftarrow "Résultat"

Variables prixHT : réel

Variable prixTTC, montantTVA : réels {déclarations}

Début {préparation du traitement}

afficher("Donnez-moi le prix hors taxe :")

saisir(prixHT)

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

afficher(Titre) {présentation du résultat}

afficher(prixHT, «euros H.T. + TVA, devient », prixTTC, «euros T.T.C. (dont TVA) », montantTVA)

Fin

INSTRUCTIONS SÉQUENTIELLES

RÉSULTAT D'UN ALGORITHME

Constante (SEUIL : réel) $\leftarrow 13,25$

Variables valA, valB: réels
compteur : entier
mot, tom : chaînes

valA $\leftarrow 0,56$

valB \leftarrow valA

valA \leftarrow valA \times (10,5 + SEUIL)

compteur $\leftarrow 1$

compteur \leftarrow compteur + 10

mot \leftarrow " Bonjour "

tom \leftarrow "Au revoir ! "

Quelles sont les différentes valeurs des variables ?

SIMULATION D'UN ALGORITHME

Algorithme CaDoitEchanger?

{Cet algorithme}

Variables valA, valB: réels {déclarations}

Début {préparation du traitement}

Afficher ("Donnez-moi deux valeurs :")

Saisir (valA, valB)

Afficher ("Vous m'avez donné ", valA, " et ", valB)

 {traitement mystère}

 valA ← valB

 valB ← valA {présentation du résultat}

Afficher ("Maintenant, mes données sont : ", valA, " et ", valB)

Fin

Que fait cet algorithme ? Pas ce qui est prévu !

CE QU'IL MANQUE !

- Déclarer une variable supplémentaire

Variables valA, valB, valTemp: **entiers**

- Utiliser cette variable pour stocker provisoirement une des valeurs

Saisir (valA, valB)

valTemp←valA

valA←valB

valB←valTemp

STRUCTURE ALTERNATIVE

« SI ... ALORS ... SINON ... FSI » (1)

- instruction conditionnelle

si <expression logique>

| alors instructions
| [sinon instructions]

fsi

- Si l'expression logique (la condition) prend la valeur **vrai**, le premier bloc d'instructions est exécuté.
- Si elle prend la valeur **faux**, le second bloc est exécuté (s'il est présent, sinon, rien).

STRUCTURE ALTERNATIVE

« SI ... ALORS ... SINON ... FSI » (2)

- Exemple :

Algorithme SimpleOuDouble

{Cet algorithme saisit une valeur entière et affiche son double si cette donnée est inférieure à un seuil donné.)

constante (SEUIL : entier) ← 10

Variable val : entier

début

Afficher("Donnez-moi un entier : ") { saisie de la valeur entière}

Saisir(val)

si val < SEUIL { comparaison avec le seuil}

alors **Afficher** ("Voici son double :", val × 2)

sinon **Afficher** ("Voici la valeur inchangée :", val)

fsi

fin

STRUCTURE ALTERNATIVE

« SI ... ALORS ... SINON ... FSI » (3)

- Autre écriture de l'exemple :

Algorithme SimpleOuDouble

{Cet algorithme saisit une valeur entière et affiche son double si cette donnée est inférieure à un seuil donné.)

constante (SEUIL : entier) \leftarrow 10

Variable val : entier

début

Afficher("Donnez-moi un entier : ") { saisie de la valeur entière}

Saisir(val)

si val < SEUIL { comparaison avec le seuil}

alors val \leftarrow val \times 2

Fsi

Afficher ("Voici la valeur val :", val)

fin

STRUCTURES ALTERNATIVES IMBRIQUÉES

Afficher :

- "Reçu avec mention Assez Bien " si une note est supérieure ou égale à 12.
- " Reçu mention Passable" si elle est supérieure à 10 et inférieure à 12.
- "Insuffisant" dans tous les autres cas.

STRUCTURES ALTERNATIVES IMBRIQUÉES

si note ≥ 12

alors afficher("Reçu avec mention AB")

sinon **si** note > 10

alors afficher(« Reçu mention Passable")

sinon afficher("Insuffisant")

fsi

fsi

