

L'algorithmique

Abdelali Saidi

saidi.a@ucd.ac.ma

Objectifs du cours

- Notions de base et Introduction à l'algorithmique
- Notion de sous-programmes et lien avec la compilation
- Structures algorithmiques fondamentales et élaboration des algorithmes
- Implantation des algorithmes dans un langage de programmation
- Introduction au test unitaire
- Utilisation des tableaux - Utilisation des chaînes de caractères
- Algorithmes fondamentaux de recherche d'un élément, parcours, tri, ...
- Avoir une première notion des performances des algorithmes utilisés

Conditions d'évaluation

- Deux contrôles continus par élément de module (50% chacun)
- Algorithmique : 30%
- Langage de programmation C : 40%
- Bureautique : 30%
- Un module est acquis par validation si sa note est supérieure ou égale à 12 sur 20 sans qu'aucune note des éléments le composant ne soit inférieure strictement à 6 sur 20.

Partie 1 : Introduction à l'algorithmique

- 1 Matériels et logiciels
- 2 Conception de logiciels
- 3 Le calcul d'un sphère
- 4 Composants d'un algorithme

Matériels et logiciels

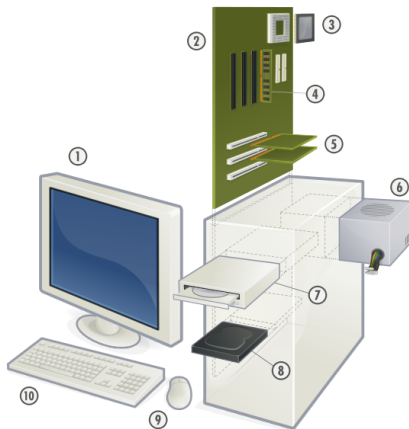


Figure: Composants matériel d'un ordinateur

Matériels et logiciels

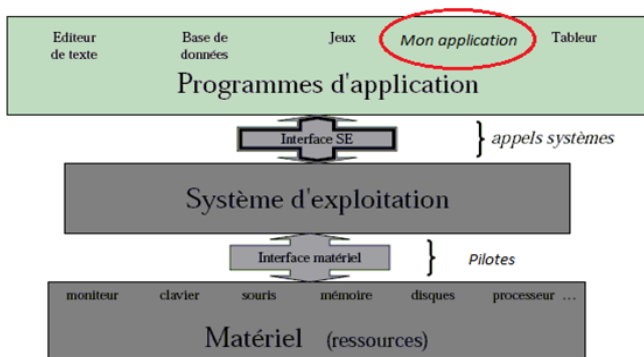


Figure: Situation du logiciel

Conception de logiciels

① Résolution du problème

- Étude des besoins (orientée clients)
- Analyse (intrants, traitement, extrants, autres informations)
- Conception de l'algorithme (architecture interne, interfaces)

② Mise en oeuvre

- Codification, programmation: traduction de l'algorithme
- Jeu de tests, test unitaires, test systèmes

③ Exploitation

- Déploiement (installation, documentation, formation)
- Maintenance (mises à jour)

Définition

Origine

L'algorithmique est un terme d'origine arabe, hommage à Al Khawarizmi (780-850) auteur d'un ouvrage décrivant des méthodes de calculs algébriques.



Définition

Définition

En informatique, un algorithme est suite finie d'opérations élémentaires constituant un schéma de calcul ou de résolution d'un problème.

Exemples d'algorithmiques

Recette de cuisine

Nous voulons préparer une recette dans la cuisine, nous aurons besoin :

- Des ingrédients et du matériels (ces sont des entrées)
- D'une méthode de préparation ou d'exécution d'opérations
- D'un certain temps

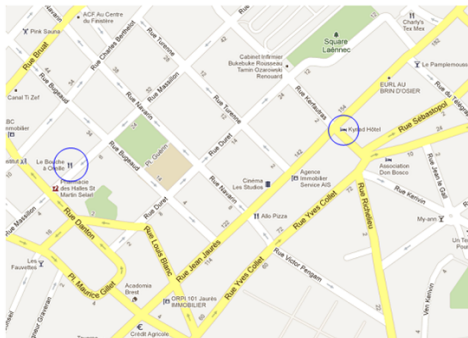
Nous aurons finalement la recette prête d'être utilisée.



Exemples d'algorithmiques

Indication routière

Méthode : Il s'agit ici de décrire une suite ordonnée d'instructions (aller tout droit, prenez la troisième à droite. . .) qui manipulent des données (carrefours, rues. . .) pour réaliser la tâche désirée (aller au restaurant).



Exemples d'opérations

Parmi les opérations que peut réaliser un algorithme :

- Lire un nombre du clavier
- Effectuer une opération mathématique
- Copier une valeur en mémoire
- Afficher une réponse à l'écran

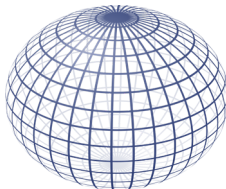
Le calcul d'un sphère

Problème :

Besoin d'un outil pour calculer le volume de sphères de dimension variable

Étapes de réalisation de cet outil de calcul :

- Analyse
- Conception
- Codification
- Mise au point



Le calcul d'un sphère

Analyse du problème

Besoin d'un outil pour calculer le volume de sphères de dimension variable

- Quelles sont les Entrée(s)?
 - Le rayon en mètre
 - Constante $\Pi = 3,1415926535897932384$
- Quelles sont les Sortie(s)?
 - Le volume en mètre cube
- Quelles sont les Opérations ?
 - Demander le rayon à l'utilisateur
 - Calculer le volume avec la formule :
 - Appliquer la formule $Volume = 4/3\Pi Rayon^3$
 - Afficher le volume

Le calcul d'un sphère

Conception de l'algorithme

- $\Pi \leftarrow 3,14159$
- Lire Rayon
- $\text{Volume} \leftarrow 4/3 * \Pi * \text{Rayon}^3$
- Écrire Volume

Le calcul d'un sphère

Conception de l'algorithme

Lors de la conception, la qualité de l'algorithme peut être discutée :

- Ex : Un logiciel rapide est couteux qu'un logiciel lent.

Il existe trois fondamentaux critères d'un bon algorithme :

- **Correct**: Il faut que le programme exécute correctement ses tâches pour lesquelles il a été conçu.
- **Complet**: Il faut que le programme considère tous les cas possibles et donne un bon résultat dans chaque cas.
- **Efficace**: Il faut que le programme exécute sa tâche avec efficacité de telle sorte qu'il se déroule en un temps minimal et qu'il consomme un minimum de ressources

Le calcul d'un sphère

Codification

Dans cette phase, l'algorithme est traduit dans un langage de programmation.

- Langage de programmation
- C, C++, Java
- Environnement de développement

Fichier résultant

- Fichiers sources
- Fichiers exécutables
- Documentation

Le calcul d'un sphère

Mise au point

Réalisation des tests pour s'assurer de la qualité du programme.

Entrée (Rayon)	Sortie (Volume)
1	4,1887
2	33,5103
0	Erreur
-1	Erreur

Composants d'un algorithme

Représentation de l'algorithme

Historiquement, deux façons pour représenter un algorithme:

- L'organigramme [Morphogramme]: représentation graphique avec des symboles (carrés, losanges, etc.)
 - + offre une vue d'ensemble de l'algorithme
 - représentation quasiment abandonnée aujourd'hui
- Le pseudo-code: représentation textuelle avec une série de conventions ressemblant à un langage de programmation (sans les problèmes de syntaxe)
 - + plus pratique pour écrire un algorithme
 - + représentation largement utilisée

Composants d'un algorithme

Exemple

Algorithme #test

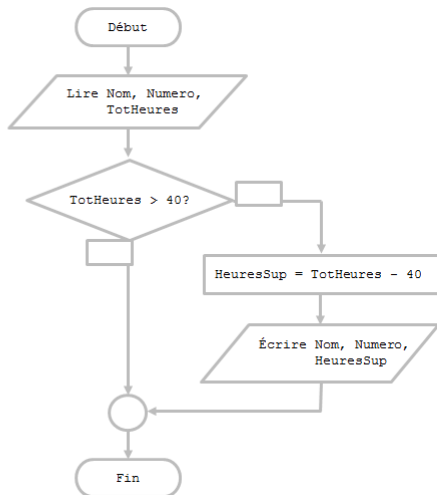
LIRE Nom, Numero, TotHeures

SI TotHeures > 40 **ALORS**

HeuresSup = TotHeures - 40

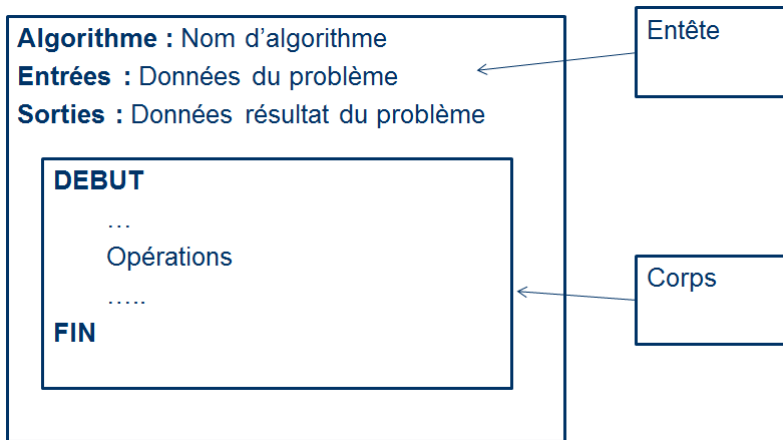
ÉCRIRE Nom, Numero, HeuresSup

FINSI



Composants d'un algorithme

Structure générale



Langages de programmation

- Pourquoi un langage de programmation ?
 - L'ordinateur ne sait exécuter qu'un certain nombre d'opérations élémentaires.
 - L'ordinateur ne comprend que le langage machine.
 - Tout type d'ordinateur a son propre langage machine
- Nécessité d'un langage commun entre le programmeur et l'ordinateur
 - Utilisable sur n'importe quel ordinateur.
 - Destiné à formuler un algorithme.
 - Comporte une syntaxe et une sémantique.
 - Exemples : C, C++, Pascal, Delphi, Fortran, Java, PHP, ...

Langages de programmation

Quel que soit le langage de programmation, il est nécessaire de le traduire en langage machine

① Compilation :

- L'ensemble du programme (écrit en langage de programmation) est traduit en langage machine.
- La suite d'instructions ainsi obtenue est exécutée par l'ordinateur.

② Interprétation :

- Les instructions (en langage de programmation) sont traduites l'une après l'autre en langage machine et exécutées directement.
- Traduction et exécution à la volée.

Langages de programmation

Algorithme

- Une suite finie d'instructions élémentaires.
- Est écrit dans un langage universel 'pseudo-code'.
- Est indépendant du langage de programmation.
- Sert à résoudre un problème informatique.



Programme

- Vient en aval de l'algorithme.
- Est écrit dans un langage de programmation.
- Résultat de la traduction de l'algorithme.
- Est indépendant de l'architecture de l'ordinateur.

Partie 2 : Concepts de variable et affectation

5 Éléments de base

6 Notion de Variable

7 Notion de constante

8 Notion d'affectation

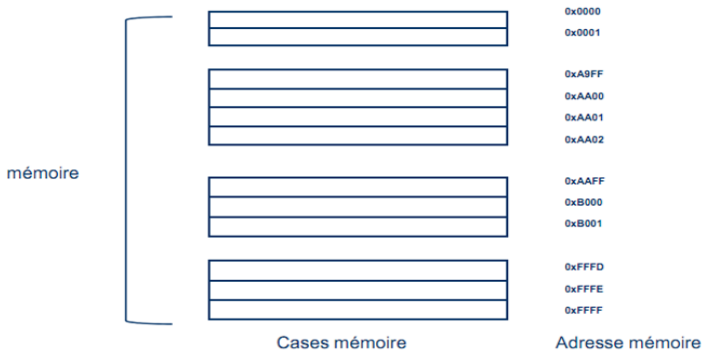
éléments de base

Un algorithme est formé de quatre types d'instructions considérées comme des petites briques de base :

- 1 l'affectation de variables
- 2 la lecture et/ou l'écriture
- 3 les tests
- 4 les boucles

Notion de Variable

- La mémoire vive stocke les logiciels en cours d'exécution
- La mémoire contient des cases mémoires stockant les données.



Notion de Variable

- Une variable désigne une case (emplacement) mémoire dont le contenu peut changer au cours d'exécution d'un algorithme/logiciel (d'où le nom de variable)
- Une variable est défini par :
 - un nom (i.e., Identificateur)
 - un type
 - une valeur

Notion de Variable

Règles de nomination

Le choix du nom d'une variable est soumis à des règles qui varient selon le langage, mais en général :

- Le nom doit commencer par une lettre alphabétique
 - Exemples : "E1" ("1E" n'est pas valide)
- Le nom doit être constitué uniquement de lettres, de chiffres et du soulignement (Éviter les caractères de ponctuation et les espaces)
 - Exemples corrects: "TEXTE2016" , "TEXTE_2016"
 - Exemples incorrects : "TEXTE 2016" , "TEXTE-2016" , "TEXTE;2016"
- Le nom doit être différent des mots réservés (par exemple en C: int, float, double, switch, case, for, main, return, ...)
- La longueur du nom doit être inférieure à la taille maximale spécifiée par le langage utilisé

Notion de Variable

Type

- Le type d'une variable détermine l'ensemble -intervalle des valeurs qu'elle peut prendre.
- Les types offerts par la plus part des langages sont :
 - ① Type numérique (entier ou réel)
 - ② Booléen
 - ③ Caractère
 - ④ Chaîne de caractère

Notion de Variable

Déclaration d'une variable

- Toute variable utilisée dans un programme doit avoir fait l'objet d'une déclaration préalable.
- En pseudo-code, la déclaration de variables est effectuée par la forme suivante :

- Variables nomVariable1, nomVariable2 : type

- Exemple :

Variables i, j, k : entier

x, y : réel

OK: booléen

Ch1, ch2 : chaîne de caractères

Notion de constante

- Une constante est une variable dont la valeur ne change pas au cours de l'exécution du programme.
- Pour déclarer une constante, on utilise la forme suivante :
 - Constante `NOM_CONSTANTE` = valeur : type,...
- Par convention les noms de constantes sont en **majuscules**

Notion de constante

Exemple

Constante $\pi = 3.14$: réel
 $\text{MAX} = 100$: entier
 $\text{MIN} = -123$: entier signé

Notion d'affectation

- L'affectation est une opération qui consiste à attribuer une valeur à une variable (c'est-à-dire remplir ou modifier le contenu d'une zone mémoire)
- En pseudo-code, l'affectation est notée par le signe \leftarrow
 - $\text{Var} \leftarrow e$: attribue la valeur de e à la variable Var
- On dit "Var reçoit la valeur de e ", "la valeur de e est affecté à Var"
- Caractéristiques :
 - e peut être une valeur, une autre variable ou une expression
 - Var et e doivent être de même type ou de types compatibles
 - l'affectation ne modifie que ce qui est à gauche de la flèche

Notion d'affectation

Exemple 1

- Soient i, j, k : entier; x : réel; ok : booléen; $ch1, ch2$: chaîne de caractères
- Que doit contenir les variables à la fin d'exécution ?

```
i ← 1
j ← i
k ← i+j
x ← 10.3
OK ← FAUX
ch1 ← « EST »
ch2 ← ch1
x ← 4
x ← j
```

Notion d'affectation

Exemple 2

- Soient i, j, k : entier; x : réel; ok : booléen; $ch1, ch2$: chaîne de caractères

```
i ← 10.3  
OK ← « VINCI »  
j ← x
```

- Ces exemples ci-dessus sont invalides

Notion d'affectation

Remarques

- Lors d'une affectation, la valeur de droite est affectée à la variable de gauche. Ainsi, $A \leftarrow B$ est différente de $B \leftarrow A$
- l'affectation est différente d'une équation mathématique :
 - Les opérations $x \leftarrow x + 1$ et $x \leftarrow x - 1$ sont différents des équations $x = x + 1$ et $x = x - 1$
 - $A + 1 \leftarrow 3$ n'est pas possible en langages de programmation et n'est pas équivalente à $A \leftarrow 2$
- Certains langages donnent des valeurs par défaut aux variables déclarées. Pour éviter tout problème il est préférable d'initialiser les variables déclarées.

Partie 3 : Expressions et opérateurs

9 Expressions

10 Opérateurs

Expressions

L'affectation

$\text{Var} \leftarrow e$: attribue la valeur de e à la variable Var

- e peut être une
 - valeur
 - variable
 - expression

Expressions

Une expression peut être une valeur, une variable ou une opération constituée de variables reliées par des opérateurs.

