

Séminaire d'algorithmique: Cours 1 et 2 Informatique Différenciée - EISTI - ING 1

Ecole Internationale des Sciences du Traitement de l'Information

Introduction

Aspect scientifique

- ▶ L'informatique est une science : premier informaticien à l'académie des sciences en 2002 : Gérard Huet

Algorithmique et informatique

- ▶ Théorie de l'information : codage et communications
- ▶ **Algorithmique** : faire vite et bien
- ▶ Programmation : écrire vite et juste

UNE définition de l'algorithmique

Art d'organiser un calcul complexe en partant d'opérations simples

Quelques illustres prédecesseurs

Euclide, Archimède, Al Khuwarizmi, Fibonacci, Hilbert, Turing, Church, Goedel, Von Neumann, Knuth, Karp, ...

Domaines d'application

- ▶ Imagerie : texture, éclairage, 3D, contours, segmentation, corrections optiques, fusion, ...
- ▶ Réseaux : gestion de trafic, diffusion, protocoles, codage
- ▶ Géométrie : trajectoires, déformations, surfaces, volumes, ...
- ▶ Nombres : éléments finis, matrices, programmation linéaire, nombres premiers, cryptage, 4 opérations arithmétiques
- ▶ Mots, textes : grammaires, automates, croisement, tri, recherche, classement, ...
- ▶ Optimisation : emploi du temps, circulation, routage, ...

Critères d'évaluation

- ▶ Machine : séquentielle, parallèle, distribuée
- ▶ **Correction**
- ▶ Temps de calcul
- ▶ Taille mémoire
- ▶ Consommation d'énergie
- ▶ Exact ou approché? Prévisible ou imprévisible? ...

Variables

Pourquoi

- ▶ Stocker temporairement des valeurs
- ▶ Données issues du disque dur, saisies par l'utilisateur, calculées par une autre partie du programme, ...
- ▶ Image d'une boîte contenant une donnée accessible via une étiquette

Comment : déclaration de variables

- ▶ Syntaxe : identifiant :type
- ▶ Exemple :

```
variables  
a: entier  
b: chaine
```

Différents types de variables (basiques)

Entier	valeur entière (ex : 12)
Réel	valeur réelle (ex : 12.34)
Booléen	valeur vrai ou faux
Chaîne	chaîne de caractère (ex : "toto")

Expressions

définition

Une expression est un ensemble de valeurs reliées par des opérateurs et équivalente à une seule valeur.

Opérateurs

Les différents opérateurs sont liés aux types des valeurs qu'ils manipulent :

- ▶ Entier : $+$, $-$, $*$, $/$, *div*, *mod*
- ▶ Réel : $+$, $-$, $*$, $/$
- ▶ Booléen : *non*, *et*, *ou*
- ▶ Chaîne : $\&$
- ▶ Divers :
 - ▶ Parenthèses (gestion des priorités) : $(,)$
 - ▶ Comparaison (résultats booléens) : $=$, \neq , $>$, $<$, \geq , \leq

En plus des parenthèses des règles de priorités implicites sont liées aux différents opérateurs.

Exemples

Considérons deux variables a et b ayant respectivement les valeurs 2 et 3

- ▶ $a * 5/2 + b$ aura pour valeur : 8
- ▶ $a * 5/(2 + b)$ aura pour valeur : 2
- ▶ "*toto*" & "*titi*" aura pour valeur : "*tototiti*"

Pourquoi

L'affectation permet d'associer une valeur à une variable

Comment : instruction d'affectation

Syntaxe : *variable* ← *expression*

Remarques

- ▶ Seule la partie gauche d'une affectation est modifiée
- ▶ L'expression qui est affectée à la variable doit être d'un type compatible avec cette variable
- ▶ La dernière affectation d'une variable écrase sa valeur précédente

Affectations

Exemples

$a \leftarrow 5$

$b \leftarrow 6$

$a \leftarrow b$

$b \leftarrow 2 * a + b$

Exercice

Comment permuter les valeurs de 2 variables ?

Exemples

$a \leftarrow 5$

$b \leftarrow 6$

$a \leftarrow b$

$b \leftarrow 2 * a + b$

Exercice

Comment permuter les valeurs de 2 variables ?

$a \leftarrow 2$

$b \leftarrow 5$

$a \leftarrow b$

$b \leftarrow a$

Exemples

$a \leftarrow 5$

$b \leftarrow 6$

$a \leftarrow b$

$b \leftarrow 2 * a + b$

Exercice

Comment permuter les valeurs de 2 variables ?

$a \leftarrow 2$

$b \leftarrow 5$

$a \leftarrow b$

$b \leftarrow a$

Pourquoi ceci n'est pas correcte ?

Exemples

$a \leftarrow 5$

$b \leftarrow 6$

$a \leftarrow b$

$b \leftarrow 2 * a + b$

Exercice

Comment permuter les valeurs de 2 variables ?

$a \leftarrow 2$

$b \leftarrow 5$

$a \leftarrow b$

$b \leftarrow a$

Pourquoi ceci n'est pas correcte ?

Comment résoudre le problème ?

Pourquoi

- ▶ Stocker une information pouvant prendre 2 valeurs (en générale opposées) : vrai ou faux
- ▶ Permettre une alternative conditionnelle lors d'une suite d'instructions (structures de contrôles)

Opérateurs

<i>ET</i>	<i>V</i>	<i>F</i>
<i>V</i>	<i>V</i>	<i>F</i>
<i>F</i>	<i>F</i>	<i>F</i>

<i>OU</i>	<i>V</i>	<i>F</i>
<i>V</i>	<i>V</i>	<i>V</i>
<i>F</i>	<i>V</i>	<i>F</i>

<i>XOR</i>	<i>V</i>	<i>F</i>
<i>V</i>	<i>F</i>	<i>V</i>
<i>F</i>	<i>V</i>	<i>F</i>

Variables booléennes

Exemple

$a \leftarrow \text{vrai}$

$b \leftarrow \text{faux}$

$c \leftarrow a \text{ ou } b$

$d \leftarrow 4$

$e \leftarrow (d = 3) \text{ ou } c$

Lecture

Entrer des valeurs externes (utilisateur) pour qu'elles soient utilisées dans la suite de l'algorithme. Implémenté en général via une saisie au clavier.

Ecriture

Communiquer des valeurs vers l'extérieur (utilisateur). Implémenté en général via un affichage à l'écran.

Syntaxe

Ecrire " Entrer une valeur"

Lire v

Ici, v est une variable contenant la valeur externe produite par l'utilisateur.

Instructions conditionnelles

Pourquoi

Les instructions conditionnelles permettent d'offrir lors du déroulement d'une suite d'instructions, une alternative selon la valeur d'un test booléen (condition).

Comment : Syntaxe

```
Si Condition Alors  
  Instructions
```

```
Fsi
```

```
Si Condition Alors  
  Instructions 1
```

```
Sinon
```

```
  Instructions 2
```

```
Fsi
```

Exemple

```
variables Temp: entier  
Ecrire "Entrez la température de l'eau"  
Lire Temp  
Si Temp <= 0 Alors  
    Ecrire "C'est de la glace"  
Sinon  
    Ecrire "C'est du liquide"  
Fsi
```

Pourquoi

Améliorer la lisibilité d'un algorithme en ajoutant des expressions logiques (booléennes) toujours vraies décrivant formellement les instructions.

Comment

Les assertions seront entourées d'accolades { et }. Elles sont toujours construites à partir de variables booléennes et de conditions élémentaires, à l'aide des opérateurs logiques *non*, *et*, *ou*

Exemple

```
variables Temp: entier  
Ecrire "Entrez la température de l'eau"  
Lire Temp  
Si Temp <= 0 Alors  
    Ecrire "C'est de la glace"  
Sinon {Temp > 0}  
    Ecrire "C'est du liquide"  
Fsi
```

Instructions conditionnelles imbriquées

Pourquoi

Il est possible d'imbruquer des instruction conditionnelles lorsque le choix dépend de plusieurs critères dépendants les un des autres.

Comment : syntaxe

```
Si Condition 1 Alors  
  Instructions 1  
Sinonsi Condition 2  
  Instructions2  
....  
Sinonsi Condition n  
  Instructions n  
Sinon  
  Instructions n+1  
Fsi
```

Instructions conditionnelles imbriquées

Exemple

```
variables Temp: entier
Ecrire "Entrez la température de l'eau"
Lire Temp
Si Temp <= 0 Alors
    Ecrire "C'est de la glace"
Sinonsi Temp < 100 {0 < Temp < 100}
    Ecrire "C'est du liquide"
Sinon {Temp >= 100}
    Ecrire "C'est de la vapeur"
Fsi
```