**Compte rendu : Algorithmique**

[1. Introduction 2](#_Toc91368493)

[2. Structures de données 2](#_Toc91368494)

[1. Variables 2](#_Toc91368495)

[2. Type de base 2](#_Toc91368496)

[3. Structure de contrôle 3](#_Toc91368497)

[1. Séquence 3](#_Toc91368498)

[2. Alternatives 3](#_Toc91368499)

[3. Itérative 4](#_Toc91368500)

[4. Applications 4](#_Toc91368501)

# 

# 1. Introduction

* L'algorithme consiste à établir une suite d’action à effectuer pour résoudre un problème (mathématique) dans un langage humain avant de passer au langage de programmation. Cela doit se faire selon une convention à respecter. La conception de l’algorithme doit être en adéquation avec le langage de programmation choisi. S’adapter aux contraintes du langage et chaque langage ne se valent pas en fonction du problème.
* La programmation ou implémentation c’est la traduction de la solution algorithmique dans un langage de programmation.
* Il y a deux catégories de langage : le langage compilé (avec génération de fichiers binaire) et le langage interprété (les fichiers sources sont directement interprétés et exécutés par ex : JavaScript, php, perl,... ).
* “=” affectation
* “==” égalité
* { et } sont des balises de début et fin
* Les mots réservés tels que: lire, écrire, début, fin,... ne peuvent pas être affectés à un variable.

# Structures de données

## 1. Variables

* une structure de données ou une variable est un moyen de stocker une donnée pour faire des traitements dessus. En général sa valeur varie. En informatique c’est un triplet: il a un nom (chaîne alphanumérique), un type (entier, car, booléen, réel) et une valeur. → {var NomDeVariable : Type; }
* Les expressions sont constituées de valeurs, de variables déclarées, des parenthèses et des opérations.
* l’affectation permet de stocker une valeur dans un variable.

## 2. Type de base

* Booléen prend comme valeur VRAI ou FAUX: résultat d’une comparaison.
* Entier (ensemble entier positif ou négatif), réel (ensemble du réel).
* Caractère ou chaîne de caractère: variable de type texte et les valeurs entre guillemets. (Attention: “1” caractère, 1 entier et 1. réel)
* Comparaison: se fait avec les opérateurs (<, \_<, ==, !=, >, \_>). comparaison de tout type de valeur (entier, réel, car et booléen). Le résultat d’une comparaison a une valeur booléenne.

# 3. Structure de contrôle

## 1. Séquence

* exécution dans l’ordre

-Faire traitement 1;

-Faire traitement 2;

- …;

-Faire traitement N;

## 2. Alternatives

* Alternative simple: Si  → instruction exécutée que si condition vaut vraie

SI (Condition) Alors

Instruction 1

Instruction 2

…

Instruction N

FSI

* Alternative simple: Si … Sinon

SI (Condition) Alors

Bloc Instructions 1

SINON

Bloc Instructions 2

FSI

* Le switch: c’est pour choisir un traitement parmi plusieurs en fonction de la valeur d’une variable entière.

SELON QUE (variableDeChoix)

CAS val 1:

Bloc instruction 1

Break;

CAS val 2:

Bloc instruction 2

Break;

CAS val 3:

Bloc instruction 3

Break;

…

Autrement: Bloc instructions

FIN SELON QUE

## 3. Itérative

Boucles ou Loop

Structure de contrôle qui permet d’éviter d’écrire la même instruction plusieurs fois lorsqu’on a des variables différents

* Pour … Faire … FinPour

-Le compteur s’incrémente automatiquement

-On sait en avance le nombre d’itération

POUR compteur ALLANT DE 1 à 10 FAIRE

Bloc Instructions

FINPOUR

* TantQue … Faire … FinTantQue

Dépend de la condition pour exécuter le bloc instruction. Il faut que la variable impliquée dans la condition puisse changer de valeur dans le corps de la boucle. Sinon soit on entre jamais soit on ne sort jamais de la boucle.

TANTQUE (Condition) FAIRE

BLOC INSTRUCTIONS

FINTANTQUE

* Faire … TantQue

On entre au moins une fois dans la boucle.

FAIRE

BLOC INSTRUCTION

TANTQUE (CONDITION)

# 4. Applications

1. Résolution d’une équation du second degré

**Algo 1: Equation du premier degré**

var a, b, x : Réel

Début

Afficher (“entrer la valeur de a et b”)

Lire (a, b)

Selon que

Cas (a==0)

Afficher (“pas de solution. Entrer une valeur de a non nulle”)

Break

Cas (b==0)

Afficher (“pas de solution. Entrer une valeur de b non nulle”)

Break

Autrement:

x= -b/a

Afficher (“La solution est:” + x)

FinSelonQue

FIN

**Algo 2 : Calcul de factorielle de n**

var n, f: entier

Début

Afficher (“entrer la valeur de n.”)

Lire (n)

Si (n==0) Alors

Afficher (“factorielle de 0 est 1”)

Sinon

n=n+1

f=n\*(n+1)

TantQue n>1

f=f\*(n-1)

n=n-1

FinTantQue

Afficher (“factorielle de n est:” + f)

Fin Si

FIN

**Algo 3: Calcul de Montant TTC, HT et la TVA**

var montant, ht, ttc, tva : réels

var nature : booléen

Début

Afficher (“entrer le montant”)

Lire (montant)

Afficher (“est-ce TTC?”)

Lire (nature)

Si (oui)

montant = ttc

ht = ttc / 1,2

tva = 0,2 \* ht

Sinon

montant = ht

ttc = ht \* 1,2

tva = 0,2 \* ht

Fin Si

Afficher (“le montant TTC=” + ttc + “, le montant HT=” + ht + “et la TVA=” + tva)

FIN