

Algorithmique et Structures de Données Statiques

1^{ière} École Prépa. de ENSSTI de Bejaia

1

Informations pratiques

- ▶ Ahror BELAID, bejaia.etudiant@gmail.com
- ▶ Plate Forme d'Enseignement à Distance



PLATE FORME D'ENSEIGNEMENT À DISTANCE
cours, collaboration et affichage en ligne

- ▶ <http://elearning.univ-bejaia.dz/>
 - ▶ Cours/faculté SE/Département Info/Licence1/ASDS
- ▶ Clé: **ALGO19**
- ▶ **Cours ppt**, TD, TP, biblio, forum, test d'évaluation,...

▶ 2

Informations pratiques

- ▶ Évaluation : Notes Examen, TP et TD
- ▶ Langage de Programmation : PASCAL

▶ 3

Objectifs

- ▶ **‘Acquisition d’une démarche méthodologique axée sur la modularité permettant la conception et la réalisation d’un petit logiciel utilisant des objets élémentaires et des données structurées statiques (tableaux à une et à deux dimensions, chaînes de caractères, ensembles, enregistrements) ’**

▶ 4

Objectifs

Être capable, partant de l'énoncé d'un problème, de :

- ▶ procéder à son découpage modulaire
- ▶ analyser et construire séparément des différents modules (principal et secondaires).
- ▶ valider chaque module
- ▶ programmer séparément les divers algorithmes correspondants aux modules (principal et secondaires)
- ▶ confectionner un dossier technique de programmation

▶ 5

Plan du cours

- 1) Éléments de base
- 2) Présentation du formalisme algorithmique
- 3) Éléments de base du langage PASCAL
- 4) Modularité
- 5) Structures de données statiques

▶ 6

Bibliographie

- ▶ L. Baba-Ahmed et S. Hocine, Algorithmique et structures de données statiques, cours et exercices avec solutions, office des publications universitaires. Cote: 005.1/169.30
- ▶ J. Courtin et I. Kowarski, Initiation à l'algorithmique et aux structures de données, Dunod. Cote: 005.7/35.6
- ▶ Granjon, Algorithmes en Pascal et en langage C, Dunod, 2004.
- ▶ Knuth, The Art of Computer Programming (TAOCP) Vol 1.-5., Addison-Wesley, Boston.
- ▶ Cormen, Leiserson, Rivest et Stein, Algorithmique : Cours avec 957 exercices et 158 problèmes, (trad. de l'anglais) Dunod, 2010. Cote: 518.1/11.6

▶ 7

Introduction

Généralités sur le traitement de l'information

Les ordinateurs sont utilisés pour

- ▶ le traitement d'informations ;
- ▶ le stockage d'informations.

▶ 8

Introduction

La notion de programme

- ▶ Tout traitement demandé à la machine, par l'utilisateur, est effectué par l'exécution séquencée d'opérations appelées **instructions**. Une suite d'instructions est appelée un **programme**.

▶ 9

Introduction

La notion de programme

- ▶ Un **programme** est une suite d'instructions permettant à un système informatique d'exécuter une tâche donnée
- ▶ La **programmation** est une écriture d'un programme dans un langage de programmation donnée.
- ▶ **Langage de programmation** : ensemble de règles de vocabulaire et de grammaire compréhensible par un ordinateur.

▶ 10

Introduction

La notion d'algorithme

- ▶ succession finie et non ambiguë d'opérations ;
- ▶ description formelle d'un procédé de traitement qui permet, à partir d'un ensemble d'informations initiales, d'obtenir des informations déduites ;
- ▶ se termine toujours (Note : semi-algorithme).

▶ 11

Introduction

Exemple d'algorithme

Algorithme qui fait la somme de deux nombres A et B et affiche le résultat:

1. Introduire le premier nombre A
2. Introduire le deuxième nombre B
3. Faire l'addition A+B et mettre le résultat dans C
4. Afficher C

▶ 13

Introduction

Langages

- ▶ Le **langage machine**:
 - ▶ le seul langage directement compréhensible par la machine (l'ordinateur).
 - ▶ une succession de 0 et de 1 définissant des opérations précises à effectuer.
 - ▶ n'est pas utilisé directement pour la programmation.

▶ 14

Introduction

Langages

- ▶ Le premier langage utilisable pour programmer un ordinateur est **l'assembleur**.
 - ▶ dépend du processeur de la machine.
 - ▶ Ses instructions sont proches de celles du langage machine.
 - ▶ permet de réaliser que des programmes relativement simples.
- ▶ Afin de réaliser des programmes plus complexes et moins dépendants de la machine, il est nécessaire d'utiliser un **langage de programmation**. Ex. : C, C++, C#, Java, Basic, Pascal, Lisp, Prolog, Fortran, Cobol, ...

▶ 15

Introduction

Définition Algorithme

- ▶ Suite finie d'opérations élémentaires permettant de résoudre un problème donné

Historique

- ▶ Vient du mathématicien et astronome Le **Muhammad ibn Musa al-Khwarizmi**, père de l'algèbre, qui formalisa au IX^e siècle la notion d'algorithme
- ▶ Ses écrits, rédigés en langue arabe, puis traduits en latin à partir du XII^e siècle, ont permis l'introduction de l'algèbre en Europe

▶ 16

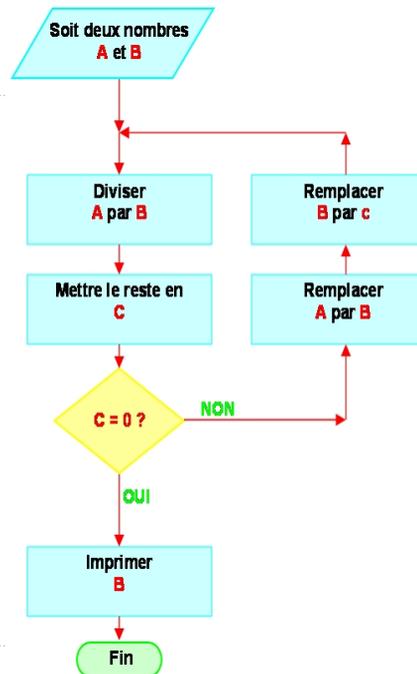
Introduction

Exemple introductif

- ▶ L'algorithme le plus célèbre est l'**algorithme d'Euclide** permettant de calculer le Plus Grand Commun Diviseur (PGCD) de deux nombres dont on ne connaît pas la factorisation

- ▶ Voir l'**organigramme**

▶ 17



Introduction

► Exemple introductif

- 1) Lire les deux entiers naturels A et B
- 2) $C \leftarrow A \text{ div } B$
- 3) Si C est différent de 0 alors
 - A \leftarrow B
 - B \leftarrow C
 - revenir au point 2
 Sinon
 - pgcd \leftarrow B
- 4) Afficher le résultat : pgcd

► 18

```

graph TD
    Start[/Soit deux nombres A et B/] --> Div[Diviser A par B]
    Div --> Rest[Mettre le reste en C]
    Rest --> Cond{C = 0 ?}
    Cond -- NON --> ReplB[Remplacer B par C]
    ReplB --> ReplA[Remplacer A par B]
    ReplA --> Div
    Cond -- OUI --> Print[Imprimer B]
    Print --> End([Fin])
  
```

Introduction

► Exemple introductif

► Résolution de l'équation du premier degré : $a \cdot x + b = 0$ dans \mathbb{R}

```

lire les coefficients a et b
si a = 0 alors
  si b = 0 alors
    afficher ("l'ensemble des solutions est R")
  sinon
    afficher ("pas de solution")
  fin de si
sinon
  solution : - b / a
  afficher ("La solution est : ", solution)
fin de si
  
```

► 19

Introduction

- ▶ Exemple introductif
- ▶ Remarque
 - ▶ Les instructions utilisées sont : lire, afficher, si ... alors ... sinon ...,
<- (affectation)
 - ▶ Les symboles a et b représentent les données de l'algorithme
 - ▶ Le symbole solution représente une variable de l'algorithme

▶ 20

Introduction

Quelques points importants :

Un algorithme :

- ▶ ne doit pas être ambigu
- ▶ doit être une combinaison d'opérations élémentaires
- ▶ doit fournir un résultat en un nombre fini d'opérations, quelles que soient les données d'entrée.

▶ 21

Introduction

Un algorithme est constitué

- ▶ d'un entête composé du **MOT Réservé ALGORITHME** et d'un nom de l'algorithme à réaliser
- ▶ d'une zone de **déclaration des identificateurs** (variables) utilisés dans l'algorithme et d'un **corps** délimité par deux mots réservés **DEBUT** et **FIN**.
C'est ici qu'on écrit les actions de l'algorithme

```
ALGORITHME <NOM>  
VAR <Déclaration des variables>  
DEBUT  
<Itérations>  
FIN
```

▶ 22