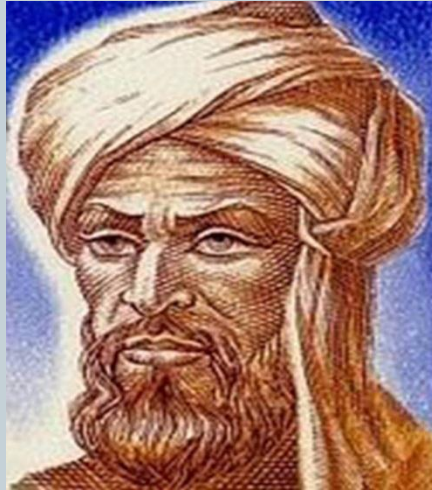


e

e

# Algorithmique

1



e

تعليم -

**YOUSSEF DAHANI**

# Définition / Historique



2

- Un algorithme est une suite finie d'opérations élémentaires constituant un schéma de calcul de traitement ou de résolution d'un problème.
- Algorithme -> Algorismus -> Alkhawarizmi

# Problématique



3

- Trouver une méthode de résolution (exacte ou approché) d'un problème donné.
- Exemple :
  - Equation 2<sup>nd</sup> degrés
  - Intégrale de Reimann.

# Algorithmes et programmes



4

- Un programme est l'implémentation dans un langage évolué d'un algorithme:
  - L'algorithme : étapes d'analyse (travaille du mathématicien).
  - Le programme : étapes de mise en œuvre (travaille du programmeur).

# Ingrédients

Nombre de personnes

— 4 +



2 cuillères à café de thé vert non parfumé



1 bouquet de menthe fraîche



25 morceaux de sucre



J'AJOUTE À MA LISTE DE COURSE

# Ustensiles



1 cuillère en bois



1 saladier

# Préparation

imprimer

TEMPS TOTAL : 20 MIN

Préparation : 10 min

Cuisson : 10 min

## Etape 1

Faire chauffer de l'eau, verser 1 petit verre d'eau bouillante sur 2 cuillères à café de thé vert.

## Etape 2

Après une minute, jeter le liquide, garder le thé

## Etape 3

(c'est pour enlever la première amertume du thé), puis verser le reste de l'eau (il faut avoir une petite théière, équipée idéalement d'un filtre dans le bec verseur, c'est + facile).

## Etape 4

Ajouter ensuite la menthe fraîche, en noyant tout de suite les feuilles dans l'eau (si elles surnagent, elles prennent un goût de brûlé, un peu amer -avec l'habitude, vous le reconnaîtrez- c'est pour les marocains quand le thé est brûlé).

## Etape 5

Par dessus, ajouter les sucres (au moins 25-30 sucres pour 4-5 verres).

## Etape 6

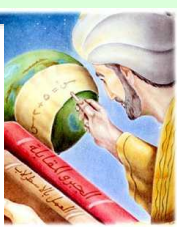
Pour mélanger, on ne remue pas avec une cuiller,

## Etape 7

on prend un verre, on verse de la théière dans le verre, puis le verre dans la théière, et ainsi de suite plusieurs fois, le contenu sera mélangé.

## Etape 8

On verse dans les verres en tenant la théière bien haut pour faire 'mousser' le sucre... et on boit très



# Structure d'un algorithme



6

- Entête : identificateur : Nom\_d\_algorithme
- Déclarations (variables, constantes, fonctions, structures ...)
- Début
  - Corps de l'algorithme
- Fin



# Conditions générales



7

- **Un algorithme doit être :**
  - **Lisible :** Compréhensible (structure, notation)
  - **De haut niveau :** indépendant d'une plate forme de développement donnée.
  - **Précis :** lever les ambiguïtés.
  - **Concis :** ne doit pas dépasser une page.
  - **Structuré.**

# Exemple



8

**Algorithme : Somme**

**Déclarations :**

**Variables :**

**a,b : réel;**

**Debut**

**lire (a) ;**

**lire (b) ;**

**Ecrire ('Somme de a et b' ,a+b) ;**

**Fin**



# Déclarations



- **Identifiant** : suite de caractères qui permet de nommer les choses
- **Variable** : une entité qui contient une information
  - Possède un identifiant
  - Possède une valeur (qui peut changer dans le temps)
  - Possède un type qui caractérise l'ensemble des valeurs
- **Une constante** : une valeur qui ne change jamais dans le temps.
- **Un type** : l'ensemble de valeur que peut prendre une variable.

# Types des données



10

- Deux grandes catégories :
  - Types simples
  - Types composés
- Pour chaque types est définit la plage des valeurs, et les opérateurs utilisés.
- Opérateur (unaire, binaire, opérande, expression ...)

# Types simples



11

- **Dénombrable :**
  - **Booleen**, les variables ne peuvent prendre que les valeurs VRAI ou FAUX
  - **Intervalle**, les variables ne peuvent prendre que les valeurs entières définies dans cet intervalle, par exemple 1..10
  - **enumere**, les variables ne peuvent prendre que les valeurs explicitees, par exemple les jours de la semaine (du lundi au dimanche), Ce sont les seuls types simples qui peuvent être définis par l'informaticien
  - **Caractere**, entoures par de simples cotes ' '

# Types simples



12

- Les types simples indénombrables:
  - **Entier** (positifs et négatifs) équivalent à  $\mathbb{Z}$
  - **Naturel** (entiers positifs) équivalent à  $\mathbb{IN}$
  - **Reel** équivalent à  $\mathbb{IR}$
  - **Chaîne de caractères**, suite de caractères entourée de doubles cotes



# Opérateurs



13

- Entier, Naturel, Réel :  
+ , - , \* , / , Div , Mod, Ent
- Booléans :  
Non, Et, Ou, OuExclusif
- Chaînes de caractères :  
+
- Enuméré :  
Succ, Pred, Ord

# Résumé



14

Type des opérandes	Opérateurs disponibles	Type résultat
Booléen	Non, Et, Ou, OuExclusif, =, ≠	Booléen
Entier, Naturel	+, -, *, div, mod, /	Entier / Réel
	=, ≠, <, >, ≥, ≤,	Booléen
Réel	+, -, *, /	Réel
	=, ≠, <, >, ≥, ≤,	Booléen
Caractère	succ, pred	Caractère
	ord	Naturel
Naturel	char	Caractère
	=, ≠, <, >, ≥, ≤,	Booléen
Chaîne de caractères	+	Chaîne
	=	Booléen



# Affectation



15

- Elle a pour but de modifier la valeur d'une variable

**IdentifVar ← NewValue (une expression)**

**Ne pas confondre l'affectation (←) avec l'opérateur d'égalité (=)**

# Entrées Sorties



16

- **Lire** : lire à partir de l'entrée standard (clavier par défaut)
  - Syntaxe : Lire(var1,var2,...)
- **Ecrire** : Afficher sur la sortie standard (Ecran par défaut)
  - Syntaxe : Ecrire(val1, val2,...)

# Exercices

17

# Implémentation sur langage évolué

18

## INTRODUCTION AU LANGAGE C

# Langages accessibles à l'ordinateur



19

- Langage machine : opération sur les bits
- Langage assembleur (symbolique) : opération sur les registres
- Langages de programmation évolués



# Le traducteur



20

- **Traducteur :**

- Compilateur sert à la traduction du programme en bloc et sert aussi à rendre le programme exécutable (indépendant de la machine).
- Interpréteur - l'exécution est liée à la machine (instruction par instruction). Il est plus lent mais interactif.



# Classification des langages



21

- **Selon le domaine d'utilisation**
  - Orientés problèmes scientifiques (Fortran & Algol)
  - Orientés problèmes de gestion (Cobol)
  - Universels (Pascal, Ada, C)
  - Orientés objet (C++, Smalltalk)
  - Pour L'IA (Prolog, Lisp)

# Classification des langages



22

- **Selon la structure interne**

- procéduraux (C, C++, Pascal, etc.) : Le programme est une suite de procédures (instructions)
- déclaratifs - logiques (Prolog) : Le programme est une suite de propositions logiques
- fonctionnels (Lisp) : Le programme est une suite de fonctions

# Classification des langages



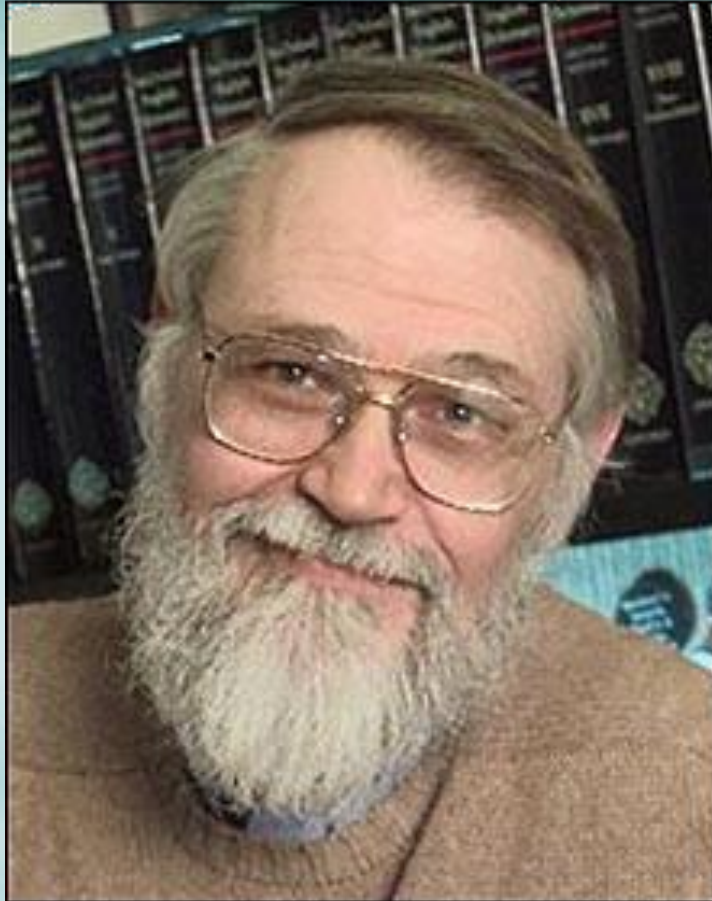
23

- **Selon la chronologie d'élaboration**
- *1<sup>ère</sup> génération* (1950-1960): Fortran, Cobol.
- *2<sup>ème</sup> génération* (1960 – 1970) : Algol, Lisp.
- *3<sup>ème</sup> génération* (1970 – 1980): Pascal, C
- 1978: *The C reference manual* (B.Kernighan, D.M. Ritchie)

# Fondateurs



24



Brian Kernighan



Dennis Ritchie



# Structure d'un programme en C



25

## Entête :

Préprocesseurs,  
Prototypes,  
déclarations globales ...

```
main() {  
    Corps du programme  
}
```

## Exemple

# Les commentaires



26

- Tous texte se trouvant entre /\* et \*/
- En C++ ou en ainsi c s'ajoute tout texte s'étalant sur une seule ligne et commençant par : //

/\* ceci est un commentaire\*/

/\* ceci est un autre ...

.... Commentaire ... \*/

// et voilà un autre commentaire



# L'entête



27

- Les
- Les



```
# include <stdio.h>
# include <stdlib.h>
//# define level 9
typedef struct disque
{
    int val;
    disque *next;
};

disque *p1 = (disque*) malloc(sizeof(disque));
disque *p2 = (disque*) malloc(sizeof(disque));
disque *p3 = (disque*) malloc(sizeof(disque));
//fonctions prototypes
disque *init();
disque *push(disque*, int);
disque *CreateTour(disque*, int);
int pop(disque**);
void hanoi(disque**,disque**,disque**, int);
void display();
```

# Le corps du programme



28

- La fonction main()

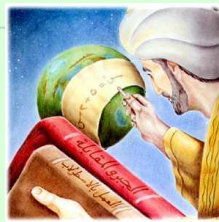
- Syntaxe

```
main ()
{
    //declaration des piles
    int level=0;
    printf("\n donner l'ordre du problème :\t");
    scanf ("%d", &level);
    p1=init (); p2=init (); p3=i
    p1=CreateTour (p1, level);
    hanoi (&p1, &p2, &p3, level);
    getchar ();getchar ();
}
```



void init (basic\_streambuf<

# Les types de données



<i>Nom de type</i>	<i>Autre Nom</i>	<i>Intervalle de valeur</i>	<i>Octets</i>
<b>int</b>	<b>signed, unsigned int</b>	<b>-32 768 à 32 767</b>	<b>2</b>
<b>short</b>	<b>short_int, signed short signed short_int</b>	<b>-32 768 à 32 767</b>	<b>2</b>
<b>long</b>	<b>long_int, signed long, signed long_int</b>	<b>-2 147 493 648 à 2 147 483 647</b>	<b>4</b>
<b>unsigned</b>	<b>unsigned int</b>	<b>0 à 65 535</b>	<b>2</b>
<b>unsigned short</b>	<b>unsigned short_int</b>	<b>0 à 65 535</b>	<b>2</b>
<b>unsigned long</b>	<b>unsigned long_int</b>	<b>0 à 4 294 967 295</b>	<b>4</b>
<b>Float</b>	<b>-</b>	<b>-3.4E38 à +3.4E38</b>	
<b>Double</b>	<b>-</b>	<b>-1.7E308 à +1.7E308</b>	
<b>long double</b>	<b>-</b>	<b>-1.7E308 à +1.7E308</b>	
<b>Char</b>	<b>signed char</b>	<b>-128 à 127</b>	<b>1</b>
<b>unsigned char</b>	<b>-</b>	<b>0 à 255</b>	<b>1</b>

# Déclaration



30

- Déclaration d'une variable.
- Déclaration des constantes.
- Etendu.
- Convention de nommage.
- Convention de valeurs.
- Déclaration multiple, initialisation.

# Séquences d'échappement



31

- `\n` = retour à la ligne.
- `\t` = tabulation.
- `\a` = alarme (un bip).
- `\0` = caractère nul (code ASCII 000);
- `\'` = l'apostrophe.
- `\\` = le *backslash*.
- `\"` = les guillemets.



# Opérateurs de base



32

- Affectation.
- Opérateurs arithmétiques.
- Affectation composée.
- Opérateurs de comparaison.
- Incrémentaux.
- Opérateurs logique.



# Entrées/Sorties standard



33

```
# include <stdio.h>
```

```
printf("Chaîne de format", arg1, arg2, ...);
```

```
scanf("Chaîne de format", &arg1, &arg2, ...);
```

# Spécificateurs de format



34

- %c** : affiche un caractère unique
- %d** ou **%i** : un entier signé sous forme décimale
- %f** : une valeur réelle avec un point décimal.
- %e** ou **%E** : une valeur réelle avec un exposant.
- %x** ou **%X** : affiche un entier hexadécimal.
- %u** : affiche un entier en notation décimale non signée.
- %s** : une chaîne de caractères (string).
- %g** ou **%G** : affiche une valeur réelle avec affichage de type e ou f selon la valeur.
- %4d** : 4 digits "au moins" réservés pour l'entier.
- %.2f** : précision de 2 rangs décimaux.