



# Introduction à l'algorithmique

Informatique et Création Numérique



## Informatique

Science du traitement **rationnel** de l'information  
par des machines automatiques

Philippe Dreyfus, 1962

# Introduction à l'algorithmique

## Informatique et Création Numérique



### Algorithme et organigramme

- Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.
- Un **organigramme** est une représentation d'une programmation sous forme d'un schéma.
- Un **programme** est une implémentation d'un algorithme ou d'un organigramme.

# Introduction à l'algorithmique

Informatique et Création Numérique



## Exemples d'algorithmes

Briques de LEGO



suite de dessins

Camion de pompiers

Meuble en kit



notice de montage

Cuisine équipée

Farine, œufs, ....



recette

gâteau

# Introduction à l'algorithmique

Informatique et Création Numérique



**Niklaus Wirth\***



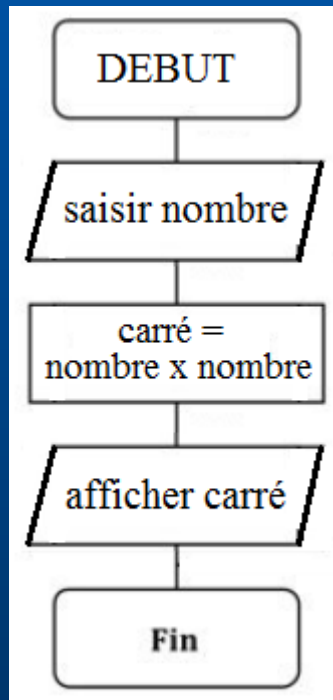
Langage **Pascal**

# Introduction à l'algorithmique

Informatique et Création Numérique



## Séquence linéaire



Algorithme EleveAuCarre

Début

Saisie

Traitement

Affichage

Fin

# Introduction à l'algorithmique

## Informatique et Création Numérique



### Un premier algorithme

Algorithme EleveAuCarre

{ Cet algorithme calcule le carré du nombre que lui fournit l'utilisateur }

Variables

unNombre, sonCarre : entier ;                    { déclarations des variables }

Début

{ préparation du traitement }

unNombre := saisir("Quel nombre voulez-vous élever au carré ?") ;

{ traitement : calcul du carré }

sonCarre := unNombre × unNombre ;

{ présentation du résultat }

afficher("Le carré est : ", sonCarre) ;

Fin

# Introduction à l'algorithmique

## Informatique et Création Numérique



### Un premier algorithme

```
program EleveAuCarre;  
{ Cet algorithme calcule le carré du nombre que lui fournit l'utilisateur }  
  
var  
  unNombre, sonCarre : integer;           { déclarations des variables }  
  
begin  
  { préparation du traitement }  
  write('Quel nombre voulez-vous élever au carré ? ');  
  readln(unNombre); { Lecture }  
  
  { traitement : calcul du carré }  
  sonCarre := unNombre * unNombre;  
  
  { présentation du résultat }  
  writeln('Le carré est : ', sonCarre);  
end.
```



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Un deuxième algorithme

Algorithme ToutOuRien

{ affiche 0 si une valeur saisie est inférieure à un seuil donné sinon affiche 1 }

constante (SEUIL : entier) := 5 ;                    { seuil à 5 V }  
variable    val : réel ;                                { valeur analogique }

début

val := saisir("Donnez un nombre :") ;

si val < SEUIL

alors

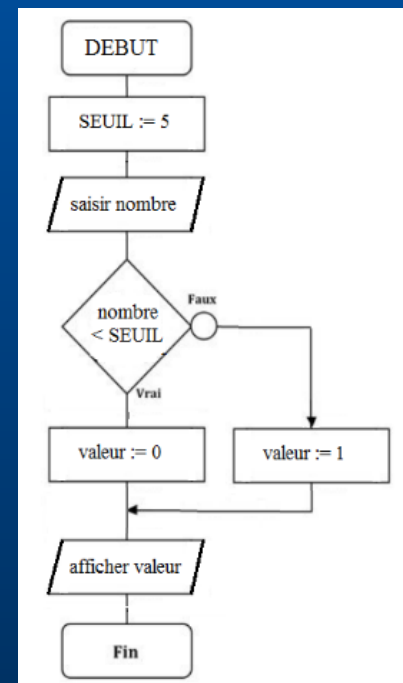
val := 0 ;

sinon

val := 1 ;

afficher("La valeur finale est : ", val) ;

fin



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Un deuxième algorithme

```
program ToutOuRien;
{ affiche 0 si une valeur saisie est inférieure à un seuil donné sinon affiche 1 }

const
  SEUIL = 5;    { seuil à 5 V }
var
  val : real;   { valeur analogique }

begin
  write('Donnez un nombre : ');
  readln(val);

  if val < SEUIL
  then
    val := 0
  else
    val := 1;

  writeln('La valeur finale est : ', val);
end.
```

# Introduction à l'algorithmique

Informatique et Création Numérique



## L'instruction conditionnelle

si <expression logique (vraie)>

alors

début

Instructions ;

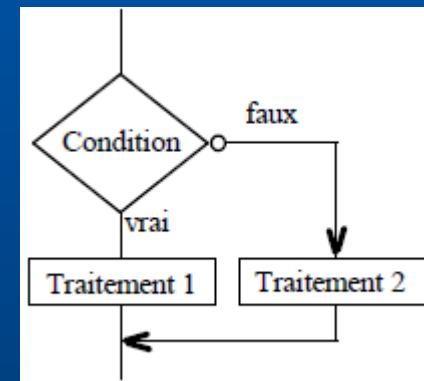
fin

sinon

début

Instructions ;

fin



# Introduction à l'algorithmique

Informatique et Création Numérique

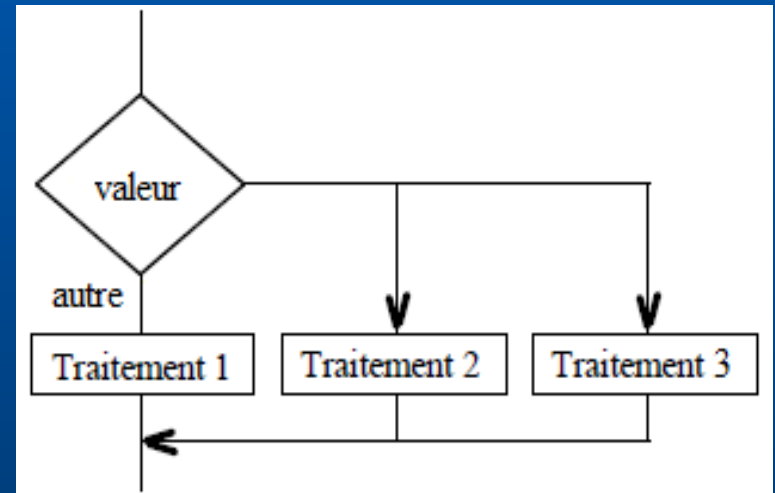


## Sélection conditionnelle

**Selon** <identificateur> **Faire**  
(Liste de) valeur :  
    Traitement() ;  
**FinFaire**

**Sinon** :  
    Traitement\_par\_defaut() ;

**finsel**



# Introduction à l'algorithmique

Informatique et Création Numérique

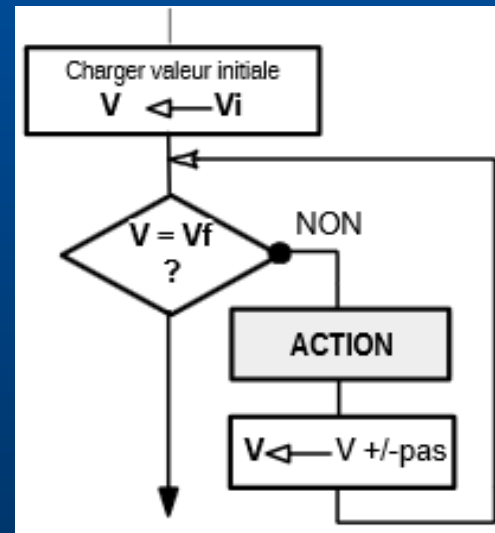


## Les instructions itératives

Pour <variable> de <initiale> à <finale> par pas de <n> Faire  
Début

Action(s)

Fin



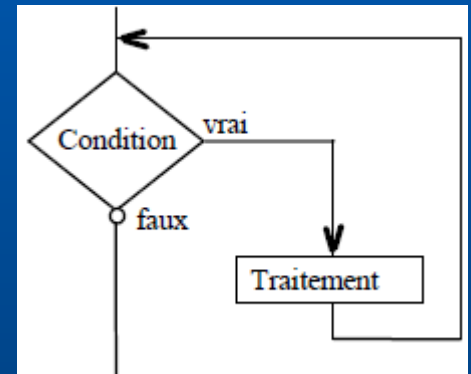
# Introduction à l'algorithmique

Informatique et Création Numérique

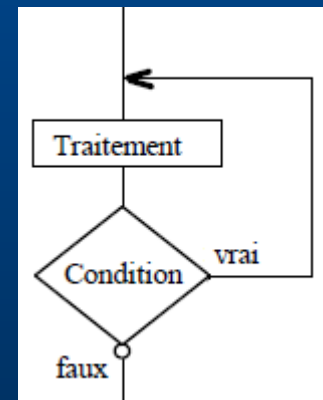


## Les instructions itératives

① **tanque** <expression logique (vraie)> **faire**  
début  
Instructions ;  
Fin



② **répéter**  
Instructions ;  
**jusqu'à** <expression logique (fausse)> ;



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Exemple 1 : le distributeur de boisson

Un distributeur propose de 2 types de boissons : eau et soda.

Le stock initial de chaque boisson est égal à 20.

Quand les stocks sont vides, le système doit avertir la maintenance et se mettre Hors Service.

Sinon, le système doit demander la boisson désirée.

- Le bouton 1 sélectionne une bouteille d'eau
- Le bouton 2 sélectionne une canette de soda

Une fois la sélection faite, si le stock de la boisson sélectionnée n'est pas vide, le système met à jour le stock, sélectionne la boisson demandée et ouvre la trappe d'accès à la boisson.

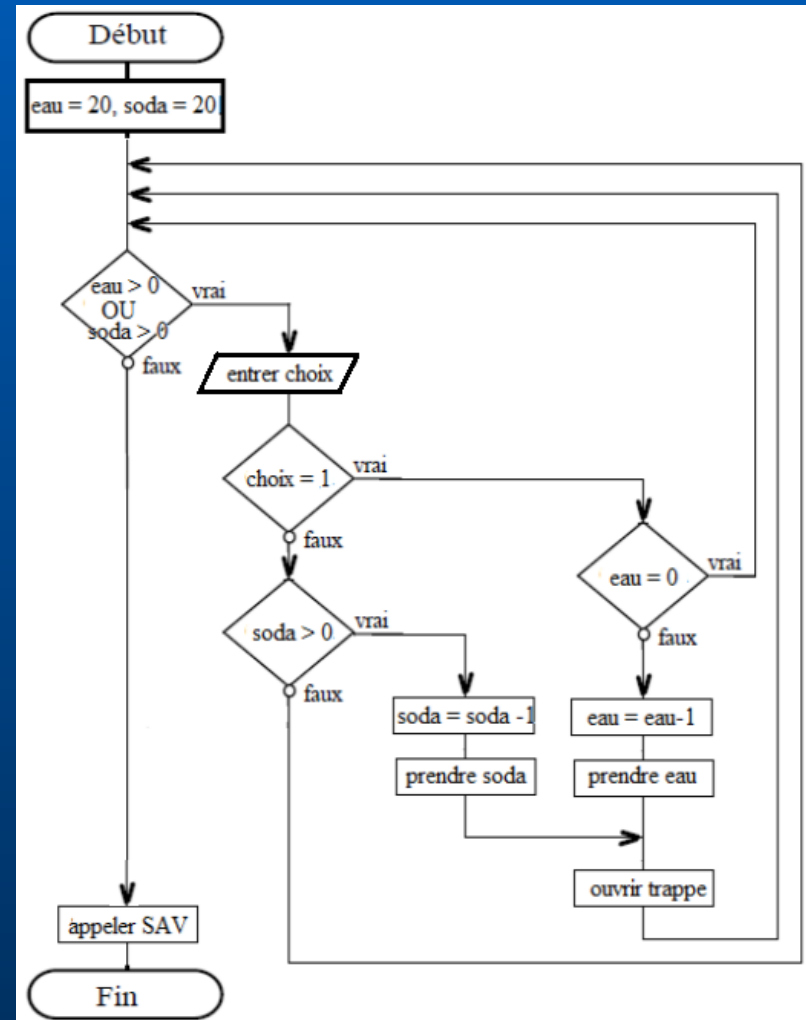
# Introduction à l'algorithmique

Informatique et Création Numérique



## Notions abordées

- Opérateur logique
- Conditions imbriquées





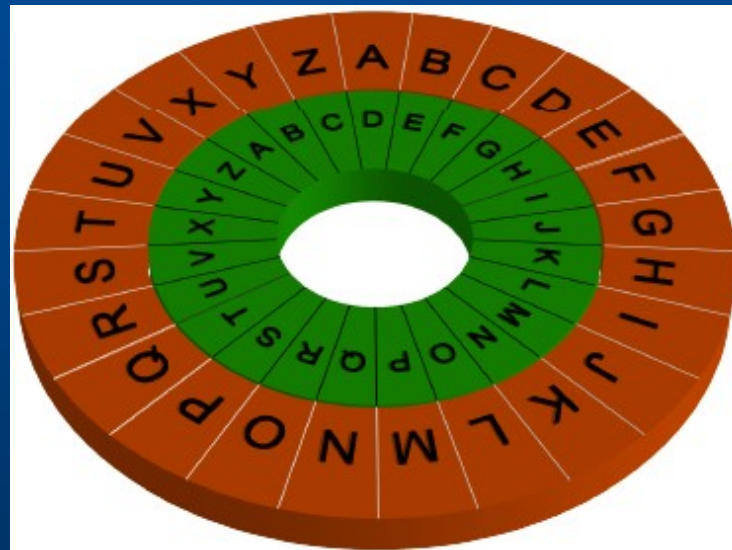
# Introduction à l'algorithmique

## Informatique et Création Numérique



### Exemple 2 : le chiffre de Caesar\*

Jules César, dans ses correspondances secrètes, codait le texte en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet.



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Le chiffre de Caesar

`mod` : modulo d'un nombre

`ord(c)` : numéro d'ordre dans la table ascii

`chr(a)` : caractère dont le code ascii est a

Les chaînes de caractères commencent à l'indice 1

# Introduction à l'algorithmique

## Informatique et Création Numérique



### Exemple 3 : le mot de passe

Réaliser l'algorithme d'un programme qui demande à un utilisateur de définir un mot de passe.

- Le mot de passe ne doit pas être inférieur à 5 caractères.
- Le mot de passe ne doit pas dépasser 10 caractères.
- Tant que le mot de passe est incorrect, on doit demander le mot de passe.
- Si le mot de passe est correct, on fait appel à un sous programme de chiffrement\*, puis on enregistre le mot de passe chiffré dans un fichier.

# Introduction à l'algorithmique

## Informatique et Création Numérique



### Notions abordées

- Fonctions et procédures
- Fichier à accès aléatoire

**assign**(f, nomf) : assigner f au fichier nomf

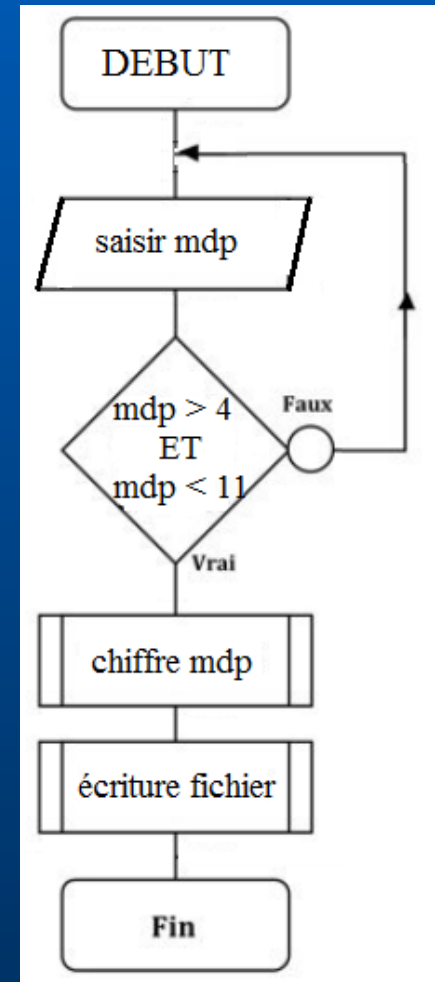
**reset**(f) : ouvrir le fichier f en lecture

**rewrite**(f) : ouvrir le fichier f en écriture

**read**(f, donnee) : lecture de données

**write**(f, donnee) : écriture de données

**close**(f) : fermer le fichier



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Exemple 4 : le tri à bulles

Mesurer la performance de l'algorithme du tri à bulles\* sur une liste de 1000 éléments.

- Créer une liste de 1000 entiers  $[0 ; 500]$  générés aléatoirement.
- Trier la liste selon l'algorithme du tri à bulles.
- Enregistrer la liste triée dans un fichier.

# Introduction à l'algorithmique

## Informatique et Création Numérique

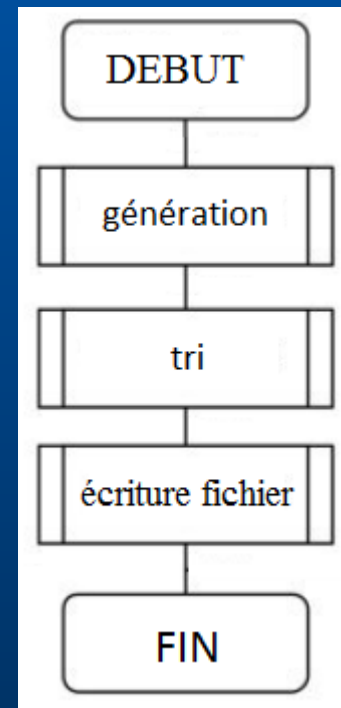
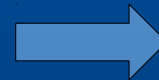


### Notions abordées

- Tableaux
- Passage par valeur et par référence

**randomize** : initialise le générateur

**random(n)** : génère un nombre  $[0..n-1]$



# Introduction à l'algorithmique

## Informatique et Création Numérique



### Exemple 5 : occurrence des lettres

Déterminer l'occurrence des lettres de l'alphabet dans un texte\*.

- Utiliser un fichier comportant au moins 1000 caractères.
- Afficher les occurrences en % par ordre décroissant.

Notion de tableaux multidimensionnels

`downto` dans boucle `for`

# Introduction à l'algorithmique

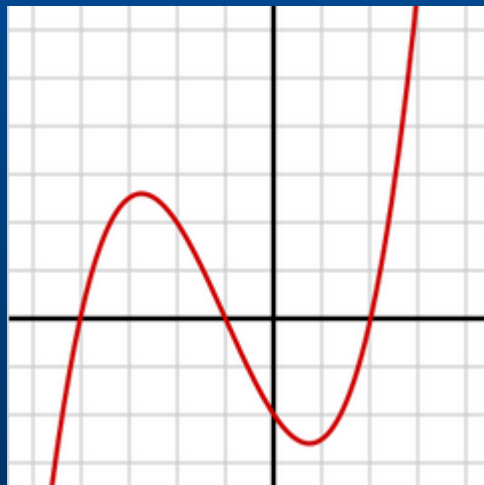
## Informatique et Création Numérique



### Exemple 6 : pour terminer...

Écrire un programme pour rechercher les racines d'un polynôme :

- Quelconque de degré 2
- D'équation :  $y = 0,25.x^3 + 0,75.x^2 - 1,5.x - 1,9$





# Introduction à l'algorithmique

Informatique et Création Numérique



## Polynôme degré 2

- Racine carrée : `sqrt()`
- Valeur absolue : `abs()`

## Polynôme degré 3

puissance : `power()`

→ écrire une fonction récursive