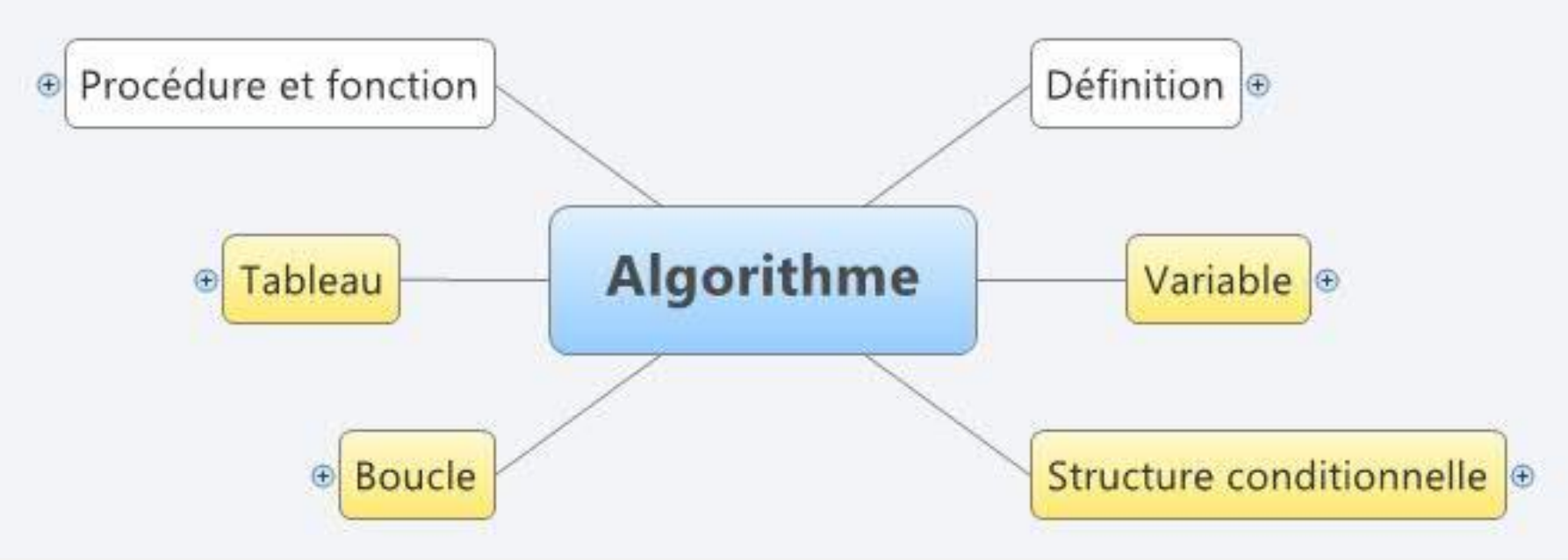
FONDAMENTAUX DE L'INFORMATIQUE

Introduction à l'algorithmique

Plan du cours



QU'EST-CE QU'UN ALGORITHME ?

Recette des pâtes au ketchup

- Ingrédients
  - Des pâtes
  - Du ketchup
  - De l'eau
- Préparation du plat
  - Faire bouillir l'eau
  - Mettre les pâtes dans l'eau
  - Attendre 7 minutes
  - Égoutter les pâtes
  - Mettre les pâtes dans l'assiette
  - Mélanger avec du ketchup
- Miam !

Recette des pâtes au ketchup

- Ingrédients
  - Des pâtes
  - Du ketchup
  - De l'eau
- Préparation du plat
  - Faire bouillir l'eau
  - Mettre les pâtes dans l'eau
  - Attendre 7 minutes
  - Égoutter les pâtes
  - Mettre les pâtes dans l'assiette
  - Mélanger avec du ketchup
- Miam !

Un but

Recette des pâtes au ketchup

- Ingrédients
  - Des pâtes
  - Du ketchup
  - De l'eau

De quoi on part ?

- Préparation du plat
  - Faire bouillir l'eau
  - Mettre les pâtes dans l'eau
  - Attendre 7 minutes
  - Égoutter les pâtes
  - Mettre les pâtes dans l'assiette
  - Mélanger avec du ketchup

- Miam !

Un but

Recette des pâtes au ketchup

- Ingrédients
  - Des pâtes
  - Du ketchup
  - De l'eau

De quoi on part ?

- Préparation du plat
  - Faire bouillir l'eau
  - Mettre les pâtes dans l'eau
  - Attendre 7 minutes
  - Égoutter les pâtes
  - Mettre les pâtes dans l'assiette
  - Mélanger avec du ketchup

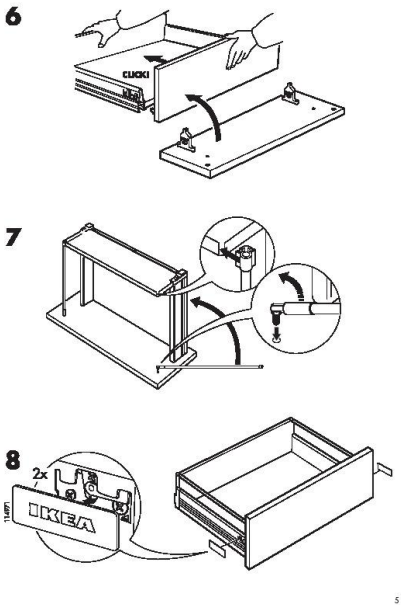
Des étapes à exécuter dans un ordre précis

- Miam !

Un but

Algorithme - exemple

Entrées	Ensemble d'instructions	Sortie



Algorithme :

George Boolos (1940–1996), philosophe et mathématicien, proposa la définition suivante :

« Des instructions explicites pour déterminer le n-ième membre d'un ensemble, pour n un entier arbitrairement grand. De telles instructions sont données de façon bien explicite, sous une forme qui puisse être utilisée par une machine à calculer ou par un humain qui est capable de transposer des opérations très élémentaires en symboles. »

Gérard Berry (1948–), chercheur en science informatique en donne la définition grand public suivante :

« Un algorithme, c'est tout simplement une façon de décrire dans ses moindres détails comment procéder pour faire quelque chose. Il se trouve que beaucoup d'actions mécaniques, toutes probablement, se prêtent bien à une telle décortication. Le but est d'évacuer la pensée du calcul, afin de le rendre exécutable par une machine numérique (ordinateur…). On ne travaille donc qu'avec un reflet numérique du système réel avec qui l'algorithme interagit. »

Algorithme :

- Le mot algorithme vient du nom arabe الخوارزمي du mathématicien perse du IXe siècle Al-Khwârizmî

- une suite d'instructions permettant de faire quelque chose

- la méthode, la façon systématique de procéder pour faire quelque chose

- le concept qui traduit la notion intuitive de procédé systématique, applicable mécaniquement, sans réfléchir, en suivant un mode d'emploi précis

- une suite finie d’opérations élémentaires constituant un schéma de calcul ou de résolution d’un problème

- une suite d'opérations élémentaires non ambiguës. Il s'achève après un nombre fini d'étapes et produit un résultat.

Algorithme :

Donald Knuth (1938–) lista les cinq propriétés suivantes comme étant les prérequis d'un algorithme :

finitude : « Un algorithme doit toujours se terminer après un nombre fini d'étapes. »
définition précise : « Chaque étape d'un algorithme doit être définie précisément, les actions à transposer doivent être spécifiées rigoureusement et sans ambiguïté pour chaque cas. »
entrées : « […] des quantités qui lui sont données avant qu'un algorithme ne commence. Ces entrées sont prises dans un ensemble d'objets spécifié. »
sorties : « […] des quantités ayant une relation spécifiée avec les entrées. »

rendement : « […] toutes les opérations que l'algorithme doit accomplir doivent être suffisamment basiques pour pouvoir être en principe réalisées dans une durée finie par un homme utilisant un papier et un crayon. »

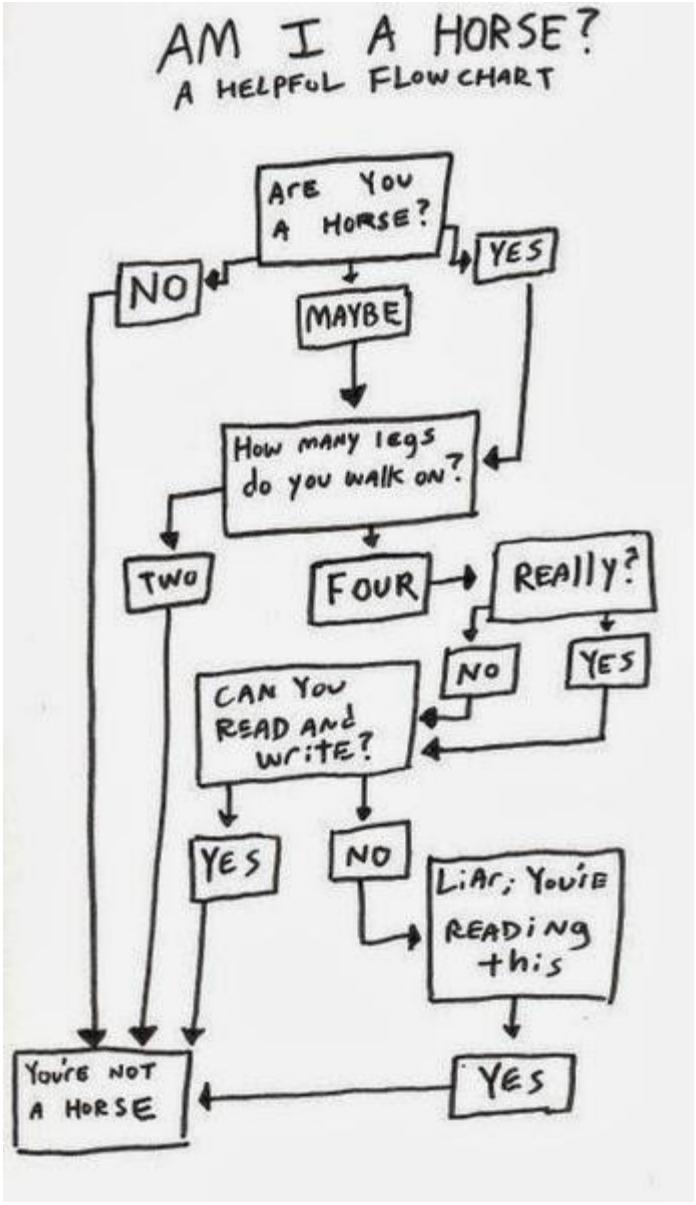
Définitions

- Un **algorithme** est une séquence d'instructions qui décrit comment résoudre un problème particulier
  - Un but, un résultat… la « **sortie** »
    - C'est souvent le seul élément dont on dispose…
  - De quoi on part… les « **entrées** »
  - Une suite d' « **instructions** », d'opérations, d'étapes à effectuer dans un ordre précis

Définitions complémentaires

A quoi sert un algorithme ?

- À décrire les étapes de résolution d'un problème :
  - de façon structurée et compacte
  - à partir d'opérations de base
  - indépendamment d'un langage de programmation
- Les données du problème en entrée
- Le résultat de sa résolution en sortie


AM I A HORSE?
A HELPFUL FLOWCHART
ARE YOU A HORSE?
NO
MAYBE
YES
HOW MANY LEGS DO YOU WALK ON?
TWO
FOUR
REALLY?
NO
YES
CAN YOU READ AND WRITE?
YES
NO
LIAR; YOU'RE READING THIS
YES
YOU'RE NOT A HORSE

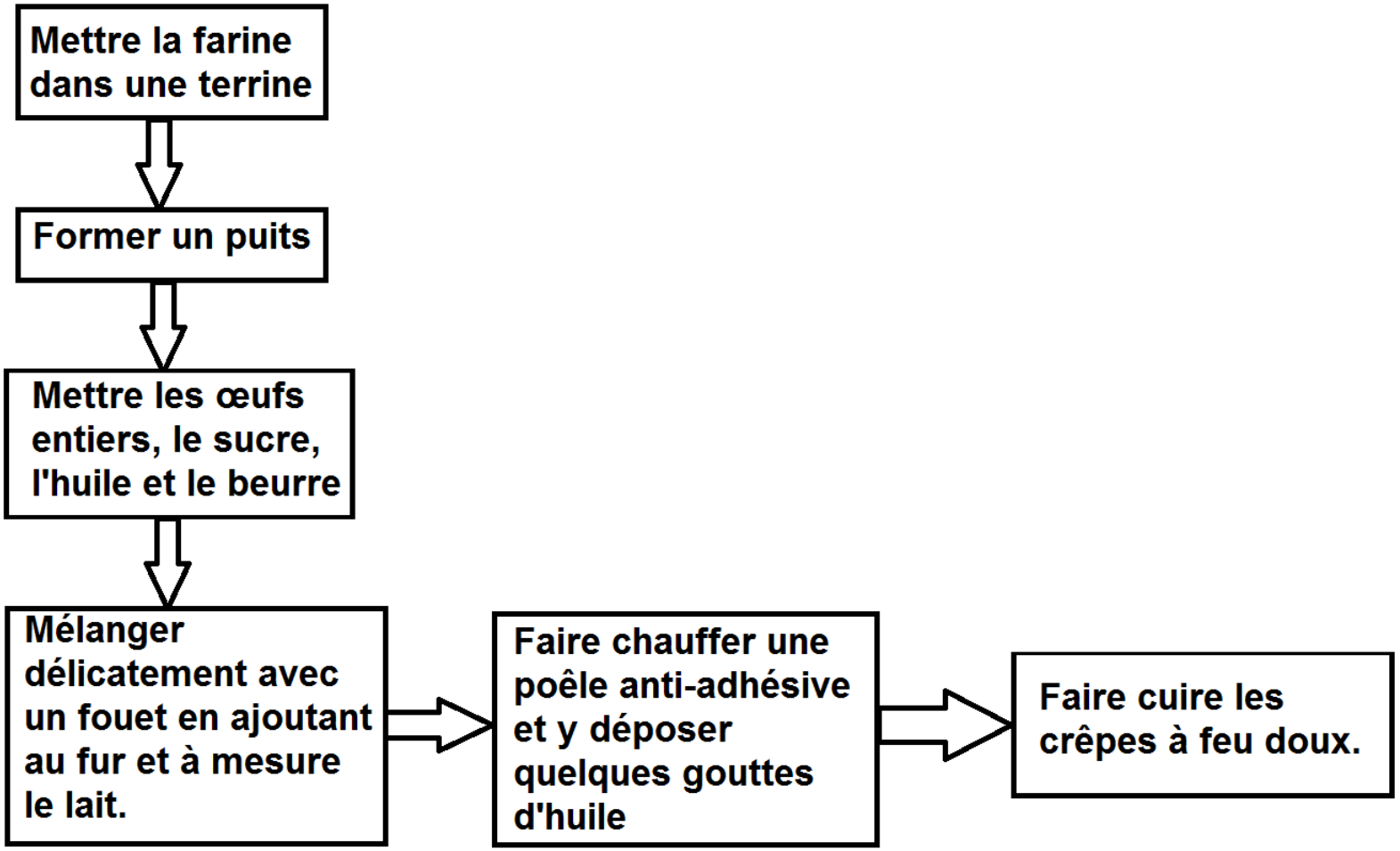
Une recette de cuisine peut être réduite à un algorithme si on peut réduire sa spécification aux éléments constitutifs :

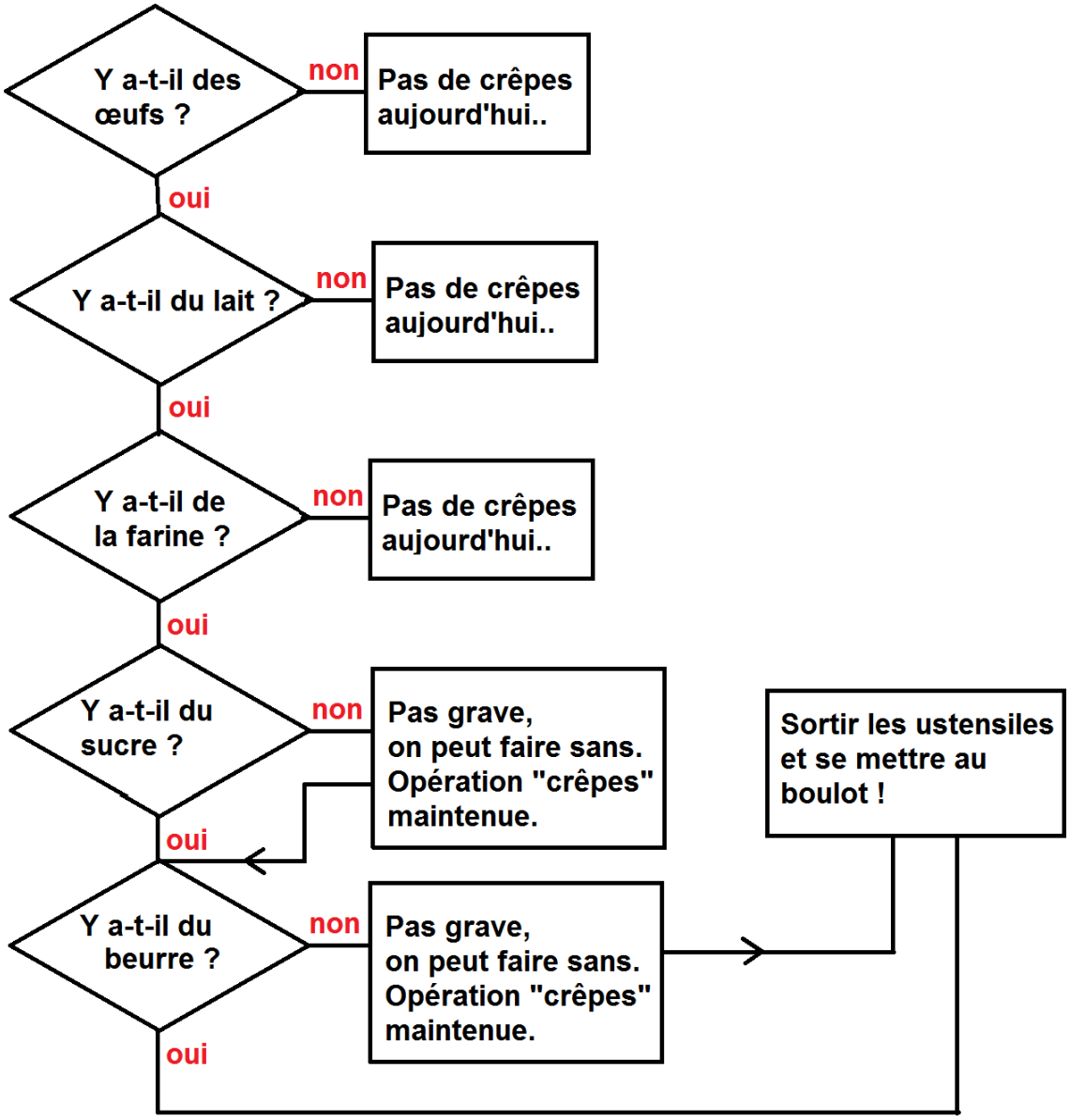
des entrées (les ingrédients, le matériel utilisé) ;

des instructions élémentaires simples (frire, flamber, rissoler, braiser, blanchir, etc.), dont les exécutions dans un ordre précis amènent au résultat voulu ;

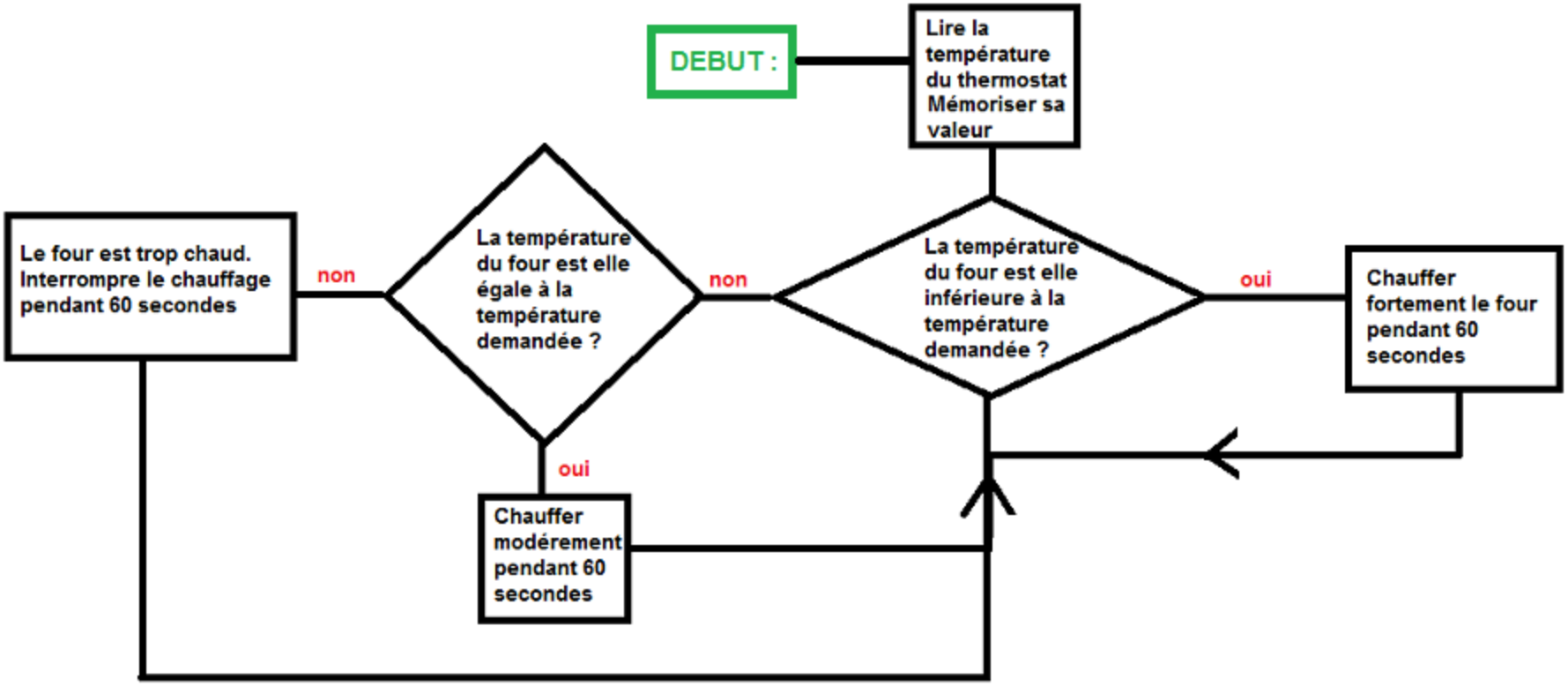
un résultat : le plat préparé.

Cependant, les recettes de cuisine ne sont en général pas présentées rigoureusement sous forme non ambiguë : il est d'usage d'y employer des termes vagues laissant une liberté d'appréciation à l'exécutant alors qu'un algorithme stricto sensu doit être précis et sans ambiguïté.


Mettre la farine dans une terrine
Former un puits
Mettre les œufs entiers, le sucre, l'huile et le beurre
Mélanger délicatement avec un fouet en ajoutant au fur et à mesure le lait.
Faire chauffer une poêle anti-adhésive et y déposer quelques gouttes d'huile
Faire cuire les crêpes à feu doux.


Y a-t-il des œufs ?
non
Pas de crêpes aujourd'hui..
oui
Y a-t-il du lait ?
non
Pas de crêpes aujourd'hui..
oui
Y a-t-il de la farine ?
non
Pas de crêpes aujourd'hui..
oui
Y a-t-il du sucre ?
non
Pas grave, on peut faire sans. Opération "crêpes" maintenue.
oui
Y a-t-il du beurre ?
non
Pas grave, on peut faire sans. Opération "crêpes" maintenue.
oui
Sortir les ustensiles et se mettre au boulot !

Algorithme bonne température :



Algorithme bonne température :

étape 1 Lire et Mémoriser Tu (température souhaitée)

étape 2 Mesurer Tf (température du four)

Si Tf est inférieure à Tu : chauffer fortement le four pendant un temps t = 60 secondes puis revenir à l'étape 2

Si Tf est égale à Tu : chauffer modérément le four pendant un temps t = 60 secondes puis revenir à l'étape 2

Si Tf est supérieure à Tu : interrompre le chauffage pendant un temps t = 60 secondes puis revenir à l'étape 2

Algorithme bonne température :

```
Début
étape 1 Lire Tu
étape 2 Lire Tf
t = 60

        Si Tf inférieur à Tu
                Alors
                exécuter compteur (t)
                                tant que t supérieur à 0
                                                exécuter chauffage fort
                                fin tant que
                aller à étape 2
        fin Si

        Si Tf = Tu
                Alors
                exécuter compteur (t)
                                tant que t supérieur à 0
                                                exécuter chauffage doux
                                fin tant que
                aller à étape 2
        fin Si

        Sinon
                exécuter compteur (t)
                                tant que t supérieur à 0
                                                exécuter chauffage stop
                                fin tant que
                aller à étape 2
        fin Si

fin
```

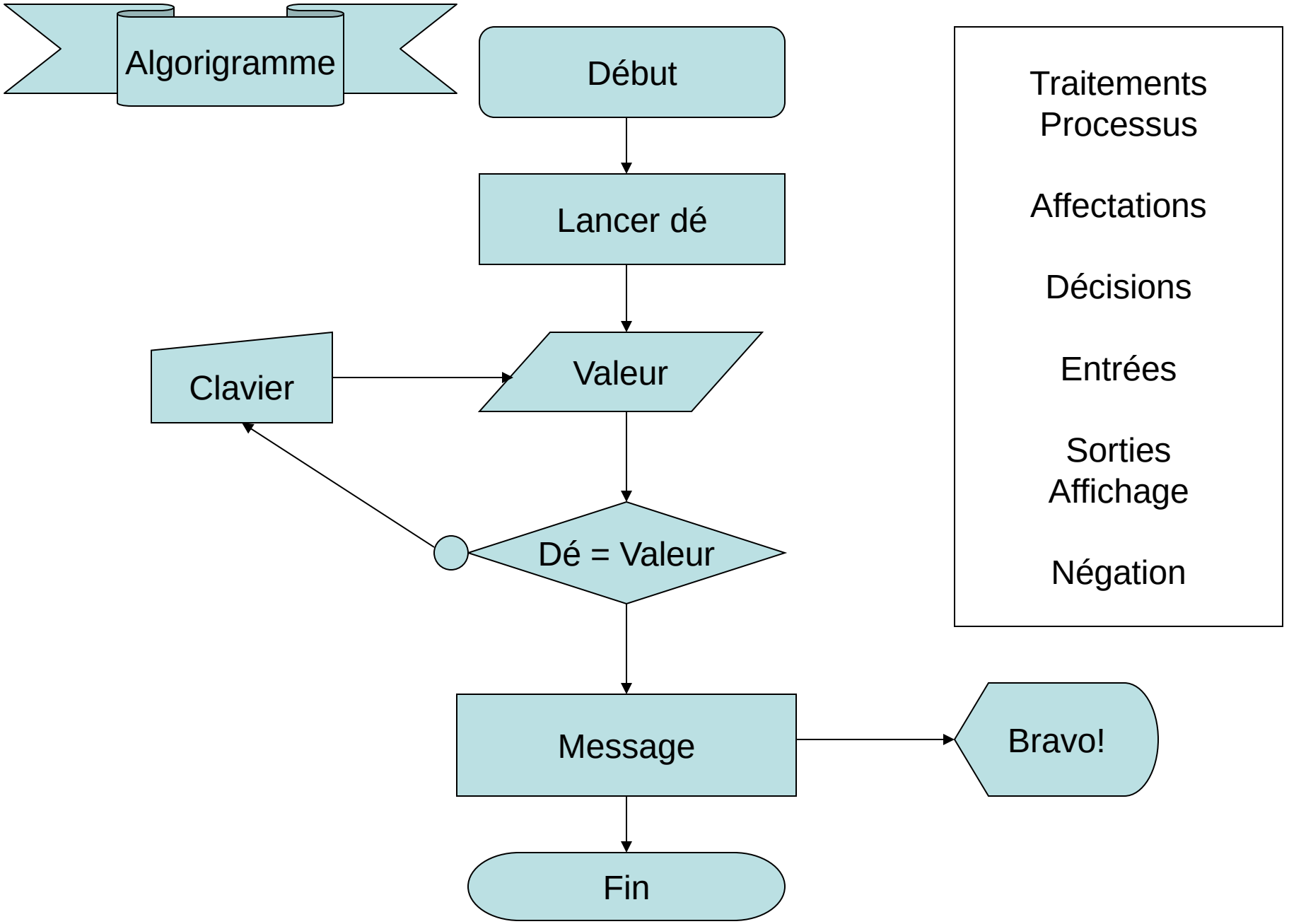
<u>Un algorithme pour simuler un lancé de dé et où une personne doit deviner la valeur</u> :

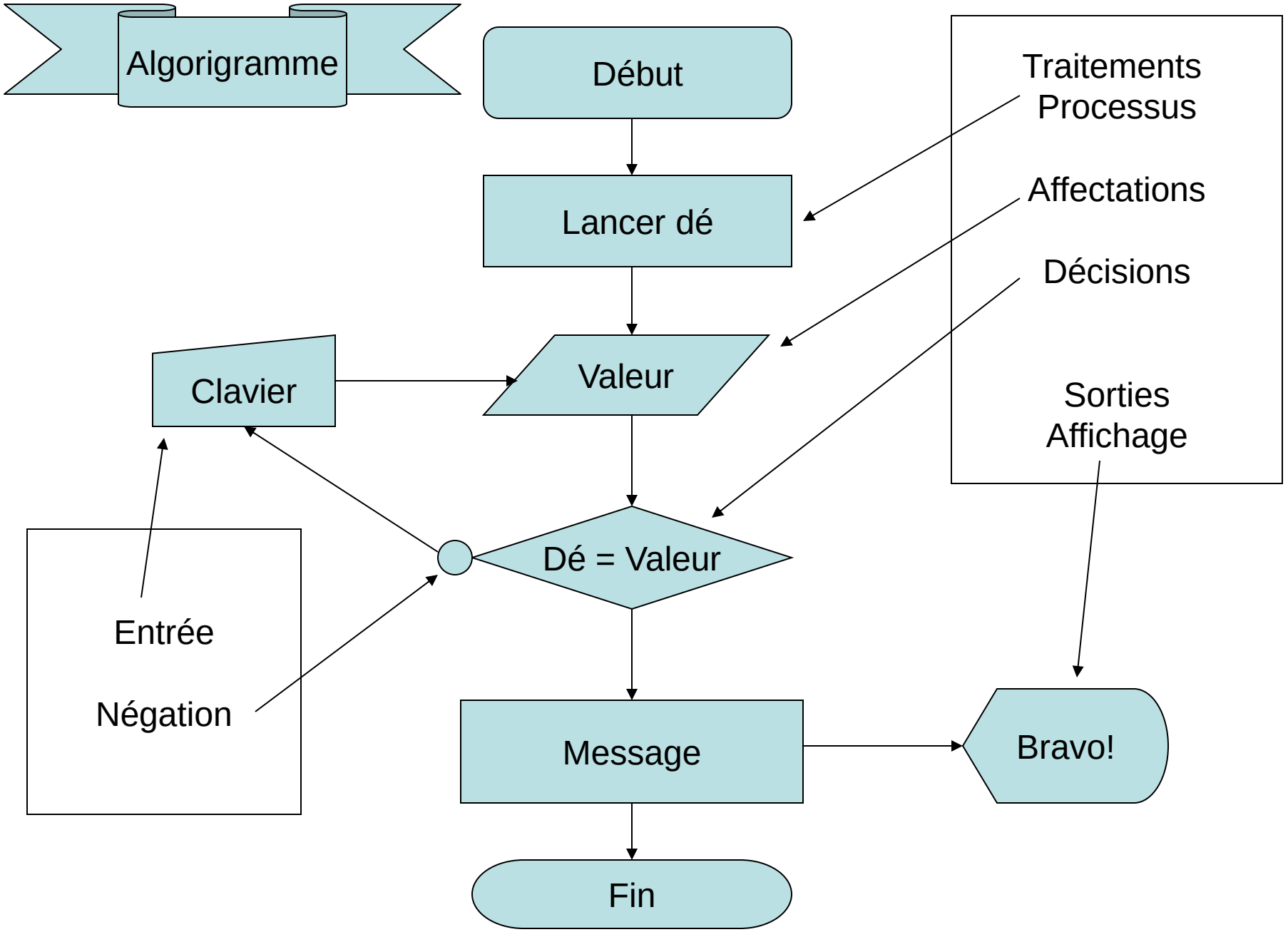
- Première étape : **lancer le dé**

- Deuxième étape : **saisir une valeur**

- Troisième étape : **si la valeur saisie est différente de la valeur du dé, retourner à la deuxième étape, sinon continuer**

- Quatrième étape : **afficher** « Bravo! »


Algorigramme
Début
Lancer dé
Clavier
Valeur
Dé = Valeur
Message
Bravo!
Fin
Traitements
Processus
Affectations
Décisions
Entrées
Sorties
Affichage
Négation


Algorigramme
Début
Traitements
Processus
Lancer dé
Affectations
Décisions
Clavier
Valeur
Sorties
Affichage
Dé = Valeur
Entrée
Négation
Message
Bravo!
Fin

```
/* Commentaire : lancer un dé, trouver sa valeur */

PROGRAMME LanceDé

/* Déclarations : variables, constantes, types, etc. */

Var
        de : entier
        valeur : entier

/* Début du programme */

DEBUT
        de ← aléatoire(6)
        valeur ← 0
        TantQue de <> valeur Faire
                Lire valeur
        FinTantQue
        Afficher « Bravo! »
FIN
```

```
#!/usr/bin/env python3
# coding: utf-8

import random

print("L’ordinateur a lancé le dé, veuillez trouver la valeur")

nbr = random.randint(1,6)

val = int(input("Veuillez saisir le numéro gagnant."))

while nbr != val:
        print("Le numéro saisi n'est pas le bon, veuillez recommencer.")
        val = int(input("Nouveau numéro"))

print("Bravo! Vous avez trouvé!")
```

```
ALGORITHME MultiplierParDeux
# Objectif : Multiplier par 2 un nombre
#            et afficher le résultat

VARIABLES
  unNombre : nombre réel
  resultat : nombre réel

DEBUT
  LIRE(unNombre)
  resultat ← unNombre * 2
  ECRIRE(resultat)
FIN
```

```
#!/bin/bash
# MultiplierParDeux
# Objectif : Multiplier par 2 un nombre
#            et afficher le résultat
# variables :
#   unNombre : nombre réel
#   resultat : nombre réel

echo "Entrez un nombre : "
read unNombre
resultat=$(( $unNombre * 2 ))
echo "le résultat est : " $resultat
```

Assembleur x86 sous DOS

```
cseg segment
assume cs:cseg, ds:cseg
org 100h
main proc
jmp debut
mess db 'Hello world!$'
debut:
mov dx, offset mess
mov ah, 9
int 21h
ret
main endp
cseg ends
end main
```

En shell Unix

```
echo "Hello world"
```

En Basic originel

```
10 Print "Hello world!"
20 End
```

En langage C

```
#include <stdio.h>
Int main(int argc, char **argv)
{
        printf("Hello world!\n");
        return 0;
}
```

En Python

```
Print ("Hello world!");
```

En Java

```
Public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

En PHP

```
<?php
        print ("Hello world!");
?>
```

En Visual Basic

```
Sub Main()
        MsgBox("Hello world!")
End Sub
```

ALGORITHMES VS SCRIPTS ET PROGRAMMES

Matériel, système d'exploitation et logiciel

Software

Scripts Logiciels

Programmes Applications

Système d'exploitation

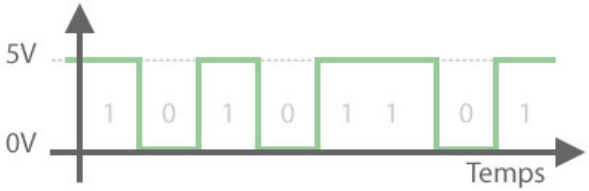
Windows Linux ANDROID iOS

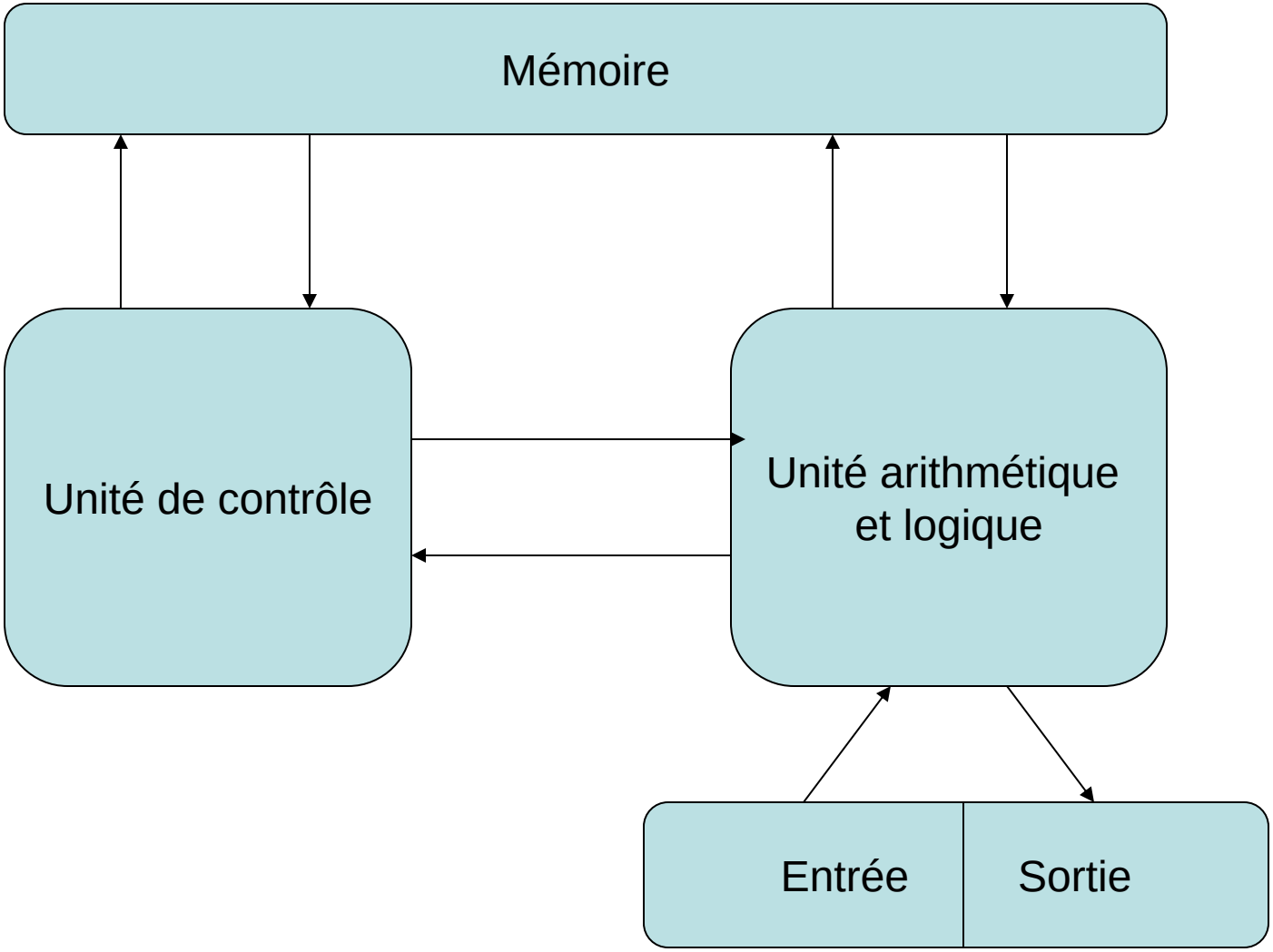
Hardware

Processeur Mémoire Carte mère Extension




0010101011101010
1010010010011111
0010101001110101
0110010010011101




Mémoire
Unité de contrôle
Unité arithmétique
et logique
Entrée
Sortie

Un ordinateur est composé principalement de :

Processeur - Unité arithmétique et logique UAL/ALU

> additions, soustractions, multiplications, divisions, modulos,
> gestion des signes (positif, négatif), opérations logiques (booléenne),
> comparaisons, rotations et décalages de valeurs, etc.

Unité de contrôle UC (CU)

Mémoire

Entrées/sorties E/S I/O

Déroulement d'un programme au sein de l'ordinateur :

- l'UC extrait une instruction de la mémoire
- analyse l'instruction
- recherche en mémoire les données concernées par l'instruction
- déclenche l'opération adéquate sur l'ALU ou l'E/S
- range le résultat dans la mémoire ou affiche sur la sortie

Le tableau ci-dessous montre la représentation des nombres de 0 à 16 dans les bases 10, 2 et 16:

	Décimal										Binaire							
Déc	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	1000	
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	Hexadécimal
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536	

Convertissons 01001101 en décimal à l'aide du schéma ci-dessous:

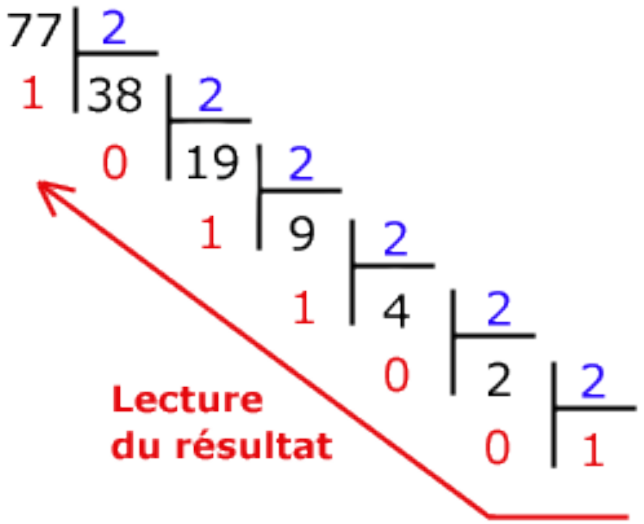
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	1	1	0	1

Le nombre en base 10 est :

$$2^6 + 2^3 + 2^2 + 2^0 = 64 + 8 + 4 + 1 = 77$$

x = hex(19) print(x) 0x13	x = bin(65) print(x) 0b1000001	x = oct(65) print(x) 0o101	x = oct(0b101101) print(x) 0o55
x = 0o10 print(x) 8	x = 0xA0F print(x) 2575	x = int(0b101010) print(x) 42	x = float(0b101010) print(x) 42.0

Allons maintenant dans l'autre sens et écrivons 77 en base 2. Il s'agit de faire une suite de divisions euclidiennes par 2. Le résultat sera la juxtaposition des restes. Le schéma ci-dessous explique la méthode :

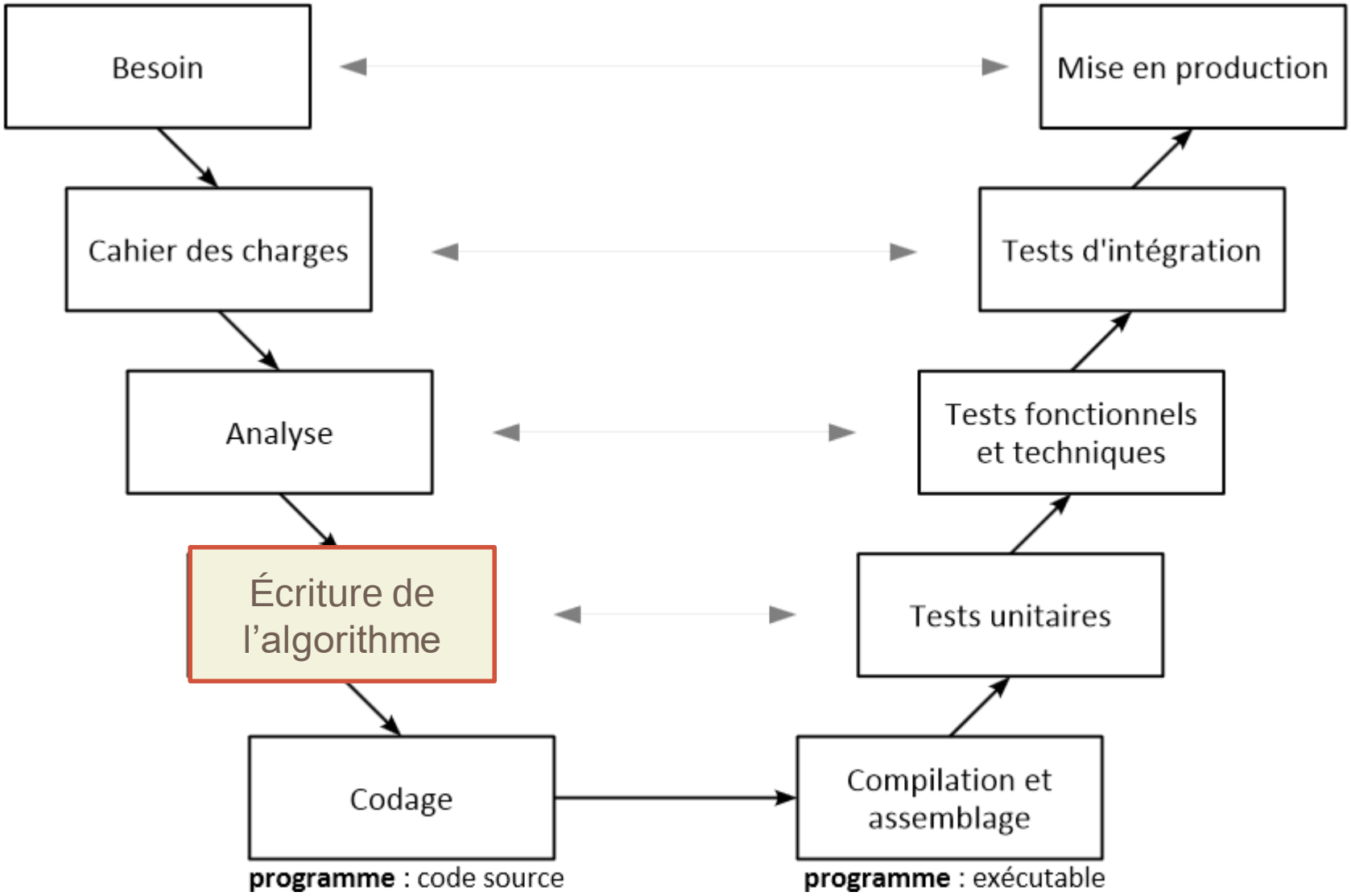


77 s'écrit donc en base 2 : 1001101

Algorithmes Vs scripts et programmes

	Algorithme	Script	Programme
Langage	**Pseudo-code** *Le pseudo-code n'est pas un langage informatique*	**Langage informatique** (bash, ksh, batch, PHP, Python, …)	**Langage informatique** (C, C++, C#, Visual Basic, java, …)
But	**Réflexion** sur une solution à un problème	**Exécution** par l'ordinateur d'une suite de commande pour produire un résultat	**Exécution** par l'ordinateur d'une suite de commande pour produire un résultat
Compréhension	Compréhension par un être humain	Pris en charge par un interpréteur de commandes	Pris en charge par des processeurs
Format global	Texte	Texte	Compilé

De l'algorithme au programme



Pour la suite de ce cours…

Pour la suite de ce cours, nous allons écrire quelques algorithmes avec un pseudo-code afin de bien travailler la partie conceptuelle de l'analyse de la problématique

Pour rendre la chose plus concrète, nous allons assez rapidement utiliser le langage de programmation « python », qui servira à exécuter nos algorithmes de manière relativement simple

Comment créer un fichier avec un code en Python ?

Depuis un ordinateur sous linux :

1. Ouvrir un éditeur de texte simple (Sous Ubuntu le Text Editor se trouve dans les Accessoires)

2. Ecrire les instructions

3. Enregistrer le fichier (peu importe le nom, par exemple multiplierParDeux.py)

4. Accorder les droits d'exécution au fichier : chmod +x multiplierParDeux.py

5. Exécuter le script dans un Terminal : python3 multiplierParDeux.py

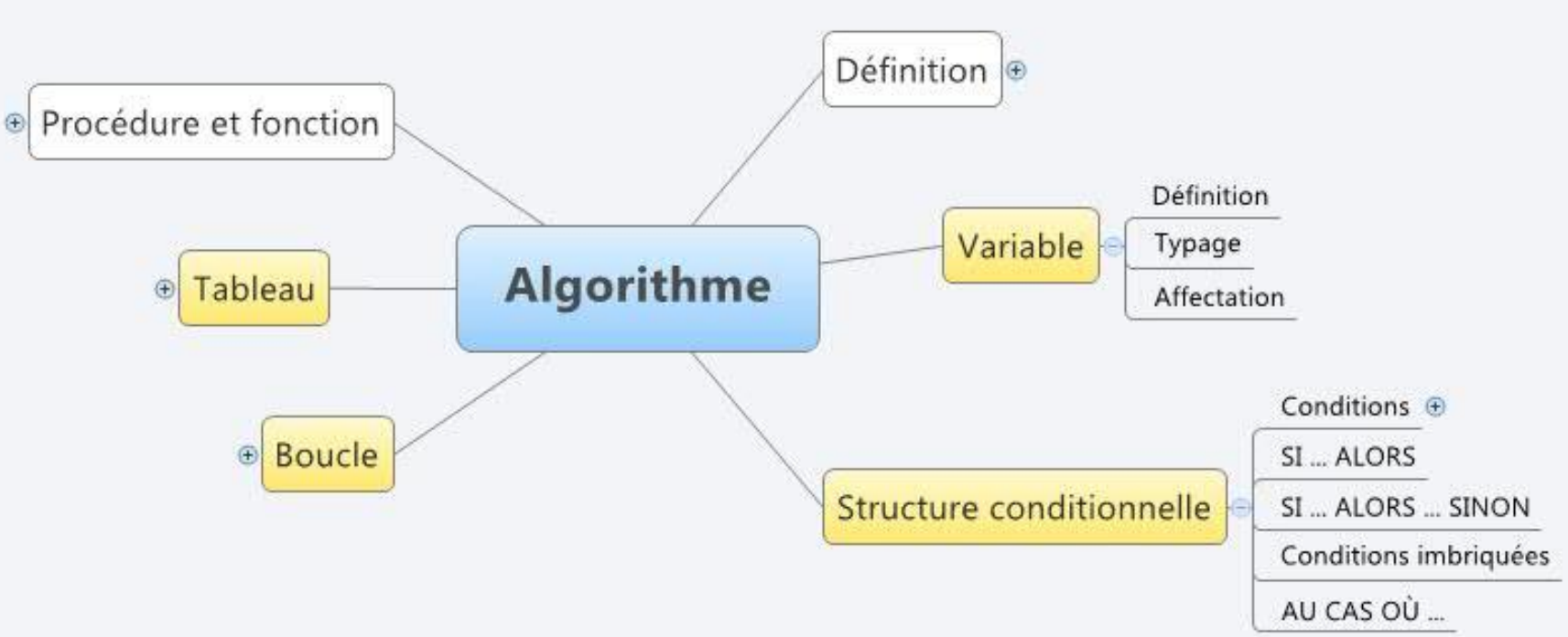
Dans le fichier multiplierParDeux.py peut se trouver le code comme celui qui suit :

```
nb = int(input("Entrez un nombre :"))
resultat = nb * 2
print("Le résultat du calcul est :", resultat)
```

Une autre alternative est :

```
print("Le résultat du calcul est :", int(input("Entrez un nombre :")) * 2)
```

LES BASES DE L'ALGORITHMIQUE


Algorithme
Définition
Procédure et fonction
Variable
Définition
Typage
Affectation
Tableau
Boucle
Structure conditionnelle
Conditions
SI ... ALORS
SI ... ALORS ... SINON
Conditions imbriquées
AU CAS OÙ ...

LES VARIABLES

Les variables - Définition

Une variable est un espace de mémoire dans lequel on peut stocker une valeur pour la réutiliser par la suite.

Autrement dit, c'est une boîte qui porte un nom et qui peut contenir quelque chose

Les variables - Exemple

```
ALGORITHME Addition
# Objectif : additionner 2 nombres
#            et afficher le résultat

VARIABLES
   nombreA : nombre réel
   nombreB : nombre réel
   resultat : nombre réel

DEBUT
   LIRE(nombreA)
   LIRE(nombreB)
   resultat ← nombreA + nombreB
   ECRIRE(resultat)
FIN
```

Les variables - syntaxe

<identificateur> : <type>

```
ALGORITHME Quelconque
VARIABLES
   nombreA : nombre réel
   nombreB : nombre entier
   description : chaine de caractères
```

Les variables – Les types

- Nombres
  - Entiers
  - Réels (nombres à virgule)
  - …
- Chaines de caractère
- Booléen (*vrai* ou *faux*)
- …

Certains langages imposent de donner un type aux variable (« typage fort »), d'autre ne le font pas (« typage faible »).

Le langage bash qui sert de support à ce cours est langage à typage faible.

Les variables - Affectation

- L'affectation est l'opération qui consiste à donner une valeur à une variable

<identificateur> ← <expression>

```
nombreA ← 1 + 8
nombreB ← nombreA - 2
```

Exercices

- Exercices sur les affectations : 1.1 à 1.6