

## 9 – CALCULABILITÉ – DÉCIDABILITÉ – RÉCURSIVITÉ - MODULARITÉ

### Compléments sur le cours de Bonnefoy & Petit

#### Points d'histoire

**1930 : thèse de Church** (Américain) sur les fonctions calculables

**1936 : machine du Britannique Turing** (table de vérité avec états + ruban mobile en lecture & écriture)

**1937 : thèse de Church-Turing** : les fonctions calculables au sens de Church sont celles qui peuvent être exécutées par une machine de Turing.

Cette thèse inspirera la fabrication du premier ordinateur et la machine de Turing sera complétée par la structure de Von Neuman.

#### Définitions à retenir (cf B.O.)

**Calculabilité** : une fonction est dite calculable si et seulement si elle renvoie le résultat demandé en un nombre fini d'étapes (ou encore si elle est exécutable par une machine de Turing).

**NB** : la calculabilité d'une fonction **ne dépend pas du langage** informatique dans laquelle elle est écrite.

**Décidabilité** : une affirmation est dite décidable si et seulement si on peut savoir, en un nombre fini d'étapes, si elle est vraie ou fausse.

**Récurtivité** : en informatique, une fonction est dite récursive si elle s'appelle elle-même.

**Module** : en Python, un module est un fichier pouvant comporter des fonctions, des variables ou des constantes, et utilisable par un programme sur fichier séparé (ex. : math, random, ...).

**Bibliothèque** (« library ») : en Python, une bibliothèque est un ensemble de modules (ex. : Tkinter, Pygame, Matplotlib, Numpy, Django, PIL, ...).

**API** (« Application Programming Interface ») : une API est une bibliothèque pouvant être utilisée par des applications écrites en différents langages (ex. : Openrouteservice : pour localiser des adresses, calculer des itinéraires).

#### Divers

**NB** : **un programme peut être une donnée d'un autre programme** (ex. : programme pris en charge par un compilateur, un débogueur, un système d'exploitation, un outil de téléchargement, ...).

**NB** : **la question de l'arrêt est indécidable** : Turing a prouvé qu'il n'existe pas d'algorithme permettant de savoir, pour n'importe quel programme, si celui-ci va s'arrêter ou non.

**Recommandation** : lorsque l'on plante un module, il convient de le **documenter** ; notamment en indiquant de quels types sont les paramètres des fonctions, ainsi que les résultats renvoyés.

**Récurtivité, modularité** : cf exercices.