

# CI 2 – ALGORITHMIQUE ET PROGRAMMATION

## TP 2 – DÉCOUVERTE DE L'ALGORITHMIQUE

### Objectifs

- Comprendre un algorithme et expliquer ce qu'il fait
- Traduire un algorithme dans un langage de programmation

*Ce document évolue. Merci de signaler toutes erreurs ou coquilles.*

### Exercice 1 – Algorithme d'Euclide

L'algorithme d'Euclide permet de calculer le PGCD de deux entiers naturels.

### Définition

#### PGCD - Plus Grand Commun Diviseur

Soient  $m$  et  $n$  deux entiers naturels (appartenant à  $\mathbb{N}$ ). On note  $\text{pgcd}(m, n)$  le plus grand entier qui divise  $m$  et  $n$ .

On donne l'algorithme permettant de calculer le PGCD .

### Pseudo Code

```

Data :  $a, b \in \mathbb{N}^*, a > b$ 
Result :  $x$ 
 $x \leftarrow a$ 
 $y \leftarrow b$ 
tant que  $y \neq 0$  faire
     $r \leftarrow$  reste de la division euclidienne de  $x$  par  $y$ 
     $x \leftarrow y$ 
     $y \leftarrow r$ 
fin
    
```

#### Question 1

*Expliquer le déroulement de cet algorithme en s'appuyant éventuellement sur plusieurs exemples.*

#### Question 2

*Implémenter ce programme en Python.*

#### Question 3

*Vérifier son fonctionnement sur plusieurs couples de nombres.*

La méthode `gcd` de la bibliothèque `fractions` permet de calculer le `pgcd` de deux nombres.

#### Question 4

Rechercher le fichier `fractions.py` sur l'ordinateur et analyser la façon dont Python calcule un pgcd. Conclure.

## Exercice 2 – Algorithme glouton – Problème du rendu de monnaie

La société Sharp commercialise des caisses automatiques utilisées par exemple dans des boulangeries. Le client glisse directement les billets ou les pièces dans la machine qui se charge de rendre automatiquement la monnaie.



#### Objectifs

Afin de satisfaire les clients, on cherche à déterminer un algorithme qui va permettre de rendre le moins de monnaie possible.

La machine dispose de billets de 20€, 10€ et 5€ ainsi que des pièces de 2€, 1€, 50, 20, 10, 5, 2 et 1 centimes.

On se propose donc de concevoir un algorithme qui demande à l'utilisateur du programme la somme totale à payer ainsi que le montant donné par l'acheteur. L'algorithme doit alors afficher quels sont les billets et les pièces à rendre par le vendeur.

#### Question 1

Pour un montant d'achat donné et pour une somme donnée par le client, proposer un algorithme en pseudo code permettant de rendre le minimum de monnaie au client. Cet algorithme devra détailler la somme à rendre (nombre de pièces et nombre de billets).

#### Question 2

Quelle structure de donnée est-il possible d'utiliser pour gérer les valeurs des billets ou des pièces ?

#### Question 3

Quel type de variable est-il préférable d'utiliser pour décompter l'argent ? Pourquoi ?

#### Question 4

Implémenter cet algorithme dans Python.

#### Question 5

Vérifier sur plusieurs cas que l'algorithme fonctionne.

## Quelques compléments sur Python

### Les Fonctions

Dans un programme, la création de fonction permet de rappeler une suite d'instruction à plusieurs reprises dans un même programme.

La fonction carrée permet de calculer le carré d'une valeur.

```
def carre(x):  
    val = x**2
```



**return** val

*Remarque : ne pas oublier l'indentation dans les instructions de la fonction.*



### Les tableaux

Les tableaux sont des collections de plusieurs éléments. Si le tableau contient  $n$  éléments, les éléments sont numérotés de 0 à  $n - 1$ .

```
>>>x=[1,"b",3,"coucou"] # Creer une liste
>>>print(x[0]) # Affichage de la premiere valeur du tableau
>>>print(x[0:2]) # Affichage des valeurs 0 a 2-1
>>>x.append(5) # Ajouter un element en fin de liste
>>>x.remove(2) # Supprime x[1]
>>>len(x) # Renvoie la taille du tableau
```



### Les dictionnaires

Les dictionnaires sont des collections de clés auxquelles sont associées des valeurs.

```
>>>dep={} # Creation d'une collection
>>>dep={"Ain":1} # On ajoute au dictionnaire la valeur 1 a la clef Ain
>>>dep["Aisne"]=2 # On ajoute au dictionnaire la valeurs 2 a la clef Aisne
>>>print(dep)
>>>print(dep['Ain'])
```