Consulter le cours à la page allophysique.com//docs/NSI/langages/page2/

Part 1

Fonctions pair / impair

On donne le script de la fonction `pair` :

```
def pair(N):
    if N==0 :
        return True
    else :
        return pair(N-2)
```

1.1 Que renvoie `pair(4)` ? Expliquez en faisant le tracé du résultat (représenter la pile d'appels).

1.2 Comment se comporte le programme si on fait : `pair(5)` ?

1.3 Modifier alors le script proposé pour tenir compte de ce problème (il faudra modifier la condition de base).

1.4 Ecrire le script de la fonction `impair`, qui renvoie True si l'argument est ...impair.

Part 2

Fonction mystère

```
def f(x,y):
  if x == y:
    return x
  elif x < y:
    return f(x,y-x)
  else:
    return f(x-y,y)
```

2.1 Sans executer le code, indiquer ce que renvoient f(5, 15) et f(8, 29)

2.2 Indiquer plus généralement ce que calcule f(x, y).

Part 3

Algorithme de calcul du PGCD d'Euclide

Voir le cours en ligne sur la complexité. L'exemple y est traité.

3.1 Historique

En mathématiques, l'algorithme d'Euclide est un algorithme qui calcule le plus grand commun diviseur (PGCD) de deux entiers, c'est-à-dire le plus grand entier qui divise les deux entiers, en laissant un reste nul.

l'algorithme d'Euclide est l'un des plus anciens algorithmes. Il est décrit dans le livre VII (Proposition 1-3) des Éléments d'Euclide (vers 300 avant JC). Cela correspond à une adaptation de la méthode naïve de calcul de la division euclidienne.

(source wikipedia)

3.2 Principe

- Il s'agit de divisions en cascade de A par B : les résultats de l'une servent à poser la suivante :
  - Le diviseur B devient le dividende ; et
  - Le reste r1 devient le diviseur.
- Arrêt lorsque le reste de la division est nul.
- Le reste trouvé avant le reste nul est le PGCD des nombres A et B.

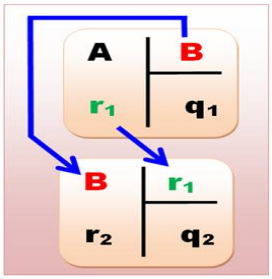

FIGURE 1 – division euclidienne

3.2.1 Compléter le tableau suivant montrant le tracé du calcul du PGCD de 264 par 228 :

division de A	par B	q	r
...			
...			
...			

3.2.2 Ecrire le script en python correspondant à la programmation itérative de cet algorithme.

```
def PGCD_it(a,b):
    """
    a et b sont des entiers, a > b
    euclide retourne un entier qui est le PGCD de a et b
    """

    ...
    ...
    ...
    ...
    ...
    ...
```

3.3 script récursif

```
def PGCD(a,b):
  """
  PGCD : entier correspondant au plus grand diviseur commun de a par b
  a et b : entiers tels que a > b
  """
  if b == 0 : return a
  else:
```

```
8       c = a % b
9       return PGCD(b,c)
```

3.3.1 Tracer le calcul de PGCD(264,228) en représentant la pile des appels successifs.

3.3.2 Prouver la terminaison de cette fonction recursive.

Part 4

La fonction de Mc Carthy

La fonction 91 de McCarthy est une fonction récursive définie par McCarthy dans son étude de propriétés de programmes récursifs, et notamment de leur vérification formelle. *(https ://fr.wikipedia.org/wiki/Fonction_91_de_McCarthy)*

C'est une fonction dont l'image est égale à 91 pour tout entier n < 102.

Elle est définie pour tout $n \in \mathbb{N}$.

$$f(n) = \begin{cases} n - 10 & \text{si} \quad n > 100 \\ f(f(n + 11)) & \text{sinon} \end{cases}$$

4.0.1 Ecrire en python le script de cette fonction recursive

4.0.2 Représenter la pile d'appels pour f(99) par cette fonction. Converge t-elle bien vers 91 ?