Devoir-Maison 5 - CORRECTION

Exercice 0.1. *Rédiger une fonction* ***FIBO(n)*** *sous Python qui :*

1. *crée le fichier* ***fibo.txt*** *où n est un entier, paramètre d'entrée de la fonction*
2. *complète ce fichier avec les n premiers termes de la suite de Fibonacci (un par ligne).*

Indication : *on pourra commencer par créer une fonction qui fabrique la liste des n premiers termes de la suite de Fibonacci.*

```
def fibo(n):
    L = [0,1] # ne fonctionne que si n>1...
    for k in range(2,n+1):
        L += [L[k-2]+L[k-1]]
    return L

def fibo_texte(n):
    L = fibo(n)
    fichier=open('fibo.txt','w') # fibo_texte(5) crée fibo.txt
    for k in L:
        mot=str(k)+'\n'
        fichier.write(mot)
    fichier.close()

n=int(input('saisir n'))
fibo_texte(n)
```

Exercice 0.2. *On cherche dans cet exercice à écrire des algorithmes permettant d'évaluer la distance, en un certain sens, entre deux listes de chiffres de même taille.*
Dans tout l'exercice, $L1$ et $L2$ désignent deux listes de même taille, notée n, dont les éléments sont des chiffres $(0, 1, 2, ..., 9)$ et qui sont notés comme en langage Python.
Par exemple, $L1[3]$ est l'élément de $L1$ de place 3). On rappelle que les deux listes $L1$ et $L2$ sont égales si et seulement si :

$$\forall i \in [\![0, n-1]\!], L1[i] = L2[i]$$

1. ***Distance discrète***
   La ***distance discrète*** *entre $L1$ et $L2$ est égale à 0 si $L1$ et $L2$ sont égales, et 1 sinon.*
   Écrire une fonction ***ddis***$(L1, L2)$ *qui, étant données deux listes de chiffres de même taille, renvoie la distance discrète associée.*
   Attention ! ! ! *On n'utilisera pas le test $L1 == L2$ pour tester l'égalité entre $L1$ et $L2$; mais on reviendra à la définition rappelée en préambule.*

```
def ddis(L1,L2):
    for i in range(len(L1)):
        if L1[i]!=L2[i]: # détection d'une valeur différente dans L1 et L2
            return 1
    return 0 # sortie de boucle ssi TOUS les tests se sont avérés faux
```

2. ***Distance de Manhattan.***
La distance de Manhattan entre $L1$ et $L2$ est le nombre :

$$\sum_{i=0}^{n-1} | L1[i] - L2[i] |$$

Écrire une fonction ***dman***$(L1, L2)$ *qui, étant données deux listes de chiffres de même taille, renvoie la distance de Manhattan associée.*
Attention ! ! ! *L'utilisation de la fonction sum est interdite.*

```
def dman(L1,L2):
    S=0
    for i in range(len(L1)): # calcul de la somme de manière itérative
        S=S+abs(L1[i]-L2[i])
    return S
```

3. ***Distance de Tchebychev***
La distance de Tchebychev entre $L1$ et $L2$ est le nombre :

$$\max_{i \in [\![0, n-1]\!]} | L1[i] - L2[i] |$$

(a) *Écrire une fonction* ***maxi***(L) *qui, étant donnée une liste L d'entiers, renvoie le maximum des éléments de ladite liste.*
Attention ! ! ! *L'utilisation de la fonction max est interdite.*

```
def maxi(L):
    m=L[0] # stockage du maximum potentiel de L
    for i in range(1,len(L)):
        if L[i]>m:
            # détection d'une valeur "trop" grande : chgt de max potentiel
            m=L[i]
    return m
```

(b) *Écrire une fonction* ***dtche***$(L1, L2)$ *qui, étant données deux listes de chiffres de même taille, renvoie la distance de Tchebychev associée.*

```python
def dtche(L1,L2):
    L=[]
    for i in range(len(L1)): # création itérative de la liste L
        L.append(abs(L1[i]-L2[i]))
    return maxi(L)
```