I11-Programmation I : Python
- TD4 -

1. Importation et utilisation de fonctions

Exercice 1. Module `math`

On rappelle une partie de la documentation du module `math`:

```
cos(...)
        cos(x)

        Return the cosine of x (measured in radians).

sin(...)
        sin(x)

        Return the sine of x (measured in radians).
```

(1) Rappeler les différentes manières d'importer ces fonctions.
(2) En important seulement les fonctions nécessaires, écrire un script qui vérifie la formule suivante pour une valeur de n saisie au clavier:

$$\cos(0)+\cos(x)+\cos(2x)+\cdots+\cos(nx)=\frac{1}{2}+\frac{\sin\left(\frac{2n+1}{2}x\right)}{2\sin(x/2)}$$

Exercice 2. Appels de fonction

On considère la fonction suivante:

```
def f(x,a,b):
    if a>b:
        a,b = b,a
    if x<=a:
        print(a)
    elif x>=b:
        print(b)
    else:
        if (x-a)>(b-x):
            print(b)
        else:
            print(a)
```

Qu'afficheront les appels suivants:
`f(0,0,0)`, `f(-2,0,3)`, `f(-2,3,0)`, `f(2,0,2)`, `f(1,0,2)`, `f(3,-1,-2)`?

Exercice 3. Permutation

On dispose d'un module `permutation` dans lequel se trouve les deux fonctions suivantes:

```
echange_car( ch, i, j)
    Echange les caractere d'indice <i> et <j> de ma chaine <ch>

remplace_car( ch, lettre):
    Remplace le premier caractere de la chaine <ch> par <lettre>
```

Écrire **une instruction** (pouvant contenir plusieurs appels de fonction imbriqués) permettant:

(1) de transformer `porte` en `sorte`
(2) de transformer `proc` en `porc`
(3) de transformer `boite` en `boire`
(4) de transformer `sauce` en `saute`
(5) de transformer `latte` en `tarte`
(6) de transformer `patte` en `passe`

2. Définition de fonctions

Exercice 4. Quelques fonctions élémentaires

(1) Écrire une fonction `SomCarreList(l)` qui retourne la somme des carrés des éléments de la liste de nombre `l`.
(2) Écrire une fonction `MoyenneList(l)` qui retourne la moyenne des éléments de la liste de nombre `l`.
(3) Écrire une fonction `NbrCar(chaine,car)` qui retourne le nombre d'occurences du caractère `car` dans la chaîne de caractères `chaine`.
(4) Écrire une procédure `SwapList(l,i,j)` qui échange les éléments d'indices `i` et `j` de la liste `l`.
(5) Écrire une fonction `EstOrdonnee(l)` qui retourne `True` si la liste de nombre `l` est ordonnée dans l'ordre croissant et `False` sinon.
(6) Écrire une fonction `MaxNeg(l)` qui retourne le plus grand entier strictement négatif d'une liste d'entiers et 0 si tous les nombres sont positifs.
(7) Écrire une fonction `Max2(l)` qui retourne les deux plus grands entiers strictement de la liste `l`.

Exercice 5. Utilisation de fonctions prédéfinies

On suppose dans tout l'exercice qu'on dispose d'une fonction `NbrDiviseurs(n)` qui retourne qui retourne le nombre de diviseurs de l'entier `n`.

(1) Écrire une fonction `EstPremier(n)` qui retourne `True` si le nombre `n` est premier, `False` sinon.
(2) Écrire une fonction `ComptePremiers(m)` qui retourne le nombre de nombres premiers inférieurs ou égaux au nombre `m`.
(3) Écrire une fonction `ListePremiers(debut,fin)` qui retourne la liste des nombres premiers compris entre `debut` et `fin`.

Exercice 6. Un peu de géometrie

Un point du plan sera représenté par un tuple de deux flottants. Notons `P` un tel tuple, ses coordonnées x et y seront donc données respectivement par `P[0]` et `P[1]`. On définit la distance entre deux points $P_1 = (x_1, y_1)$ et $P_2 = (x_2, y_2)$ par

$$d(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

(1) Écrire une fonction `Distance(P1,P2)` qui retourne la distance entre les points `P1` et `P2`.
(2) On décide de représenter un triangle par une liste de trois points. Écrire une fonction `Perimetre(T)` qui retourne le périmètre du triangle `T`.
(3) Écrire une fonction `EstEquilateral(T)` qui retourne `True` si le triangle `T` est équilatéral et `False` sinon.
(4) Écrire un script qui recherche dans une liste de triangles quelconques prédéfinie `list_triangle` le triangle équilatéral ayant le plus grand périmètre et l'affiche.

Exercice 7. Polynome

(1) Écrire une fonction `factorielle(n)` qui retourne $n! = 1 \times 2 \times 3 \times \cdots \times (n-1) \times n$.
(2) Écrire une fonction `binomial(n,k)` qui retourne la valeur du coefficient binomial $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.
(3) Écrire un script qui demande un entier n à l'utilisateur et affiche l'ensemble des coefficients du polynome $(X+1)^n = \binom{n}{0} + \binom{n}{1}X + \cdots + \binom{n}{n-1}X^{n-1} + \binom{n}{n}X^n$.