TD 2 : BOUCLES, TESTS ET LISTES

Enregistrer le fichier sous le nom `2012MA350_NOM_Prenom_AAAAMMJJ_TDN_bla` où `NOM` est votre nom de famille en majuscule et `Prenom` votre prénom. Ensuite, `AAAA` est l'année en cours (`2012` *a priori*), `MM` le mois (ici septembre, donc `09`) et `JJ` le jour (ici `24`). Enfin `N` est le numéro de TD (ici `02`) et `bla` est un texte décrivant le contenu ou dépendant de votre humeur. Le nom du fichier ne doit pas contenir de caractères spéciaux et ce, malgré leur présence dans vos nom et prénom. Exemple : `2012MA350_MARTIN_Jacques_20120924_TD2_Boucles`.

Exercice 1.

1. Que font les commandes suivantes ?

```
range (3)
range (2, 10)
range (3, 8, -1)
range (8, 3, -1)
```

2. Faire afficher à Sage la liste croissante des entiers n avec $17 \leqslant n \leqslant 50$.
3. De même avec la liste décroissante des entiers compris entre -21 et 100 tous les deux inclus avec un pas de 11 (c'est-à-dire que l'on ne veut qu'un élément sur 11).
4. Déterminer ce que fait la commande `L=[i^2 for i in range (1, 15)]; L`.
5. Écrire la liste des restes de la division euclidienne par 13 des carrés des entiers n pour $0 \leqslant n \leqslant 2011$.
6. Construire la liste des cubes des entiers positifs pairs strictement plus petits que 41.

Exercice 2.

1. Créer la liste `L` des 15 premières puissances de 2 en partant de $1 = 2^0$.
2. Déterminer ce que font les commandes suivantes `L[0]`, `L[4]`, `L[2:7]`, `L[5:]`, `L[:8]` et `L[-1]`.
3. Demander à Sage de vous rendre l'antépénultième (aussi connu sous le nom d'avant-avant-dernier) élément de `L` de deux façons différentes.
4. À partir de `L`, créer les listes `M1` et `M2` telles que
   - `M1` contient un élément de `L` sur deux, en partant de $1 = 2^0$;
   - `M2` contient un élément de `M1` sur trois, en partant de $4 = 2^2$.

Exercice 3. Une fonction se définit à l'aide de la commande `def`.

1. Que fait la fonction suivante ?

```
def factorielle (n):
    return prod (range (1, n+1))
```

2. À l'aide de `sum`, écrire la fonction `sumint (n)` qui renvoie la somme des entiers k tels que $0 \leqslant k \leqslant n$.
3. De même, écrire la fonction `sumcubes (n)` qui renvoie la somme des cubes des entiers k tels que $0 \leqslant k \leqslant n$.
4. Enfin, créer la fonction `diff (n)` qui renvoie la différence entre `sumcubes (n)` et le carré de `summint (n)`.
5. Tester `diff` sur quelques entiers, que remarque-t-on ?

Exercice 4.

1. Déterminer ce que fait la fonction suivante, en essayant pour des x relativement petits (compris entre 1 et 10)

```
def harm (x):
    n=1
    S=1.0
    while S <= x:
        n=n+1
        S=S+1.0/n
    return (n-1, N(S-1.0/n, digits=100))
```

2. Écrire la fonction `invquad (x)` qui renvoie le plus grand n tel que $\sum_{k=1}^{n} \frac{1}{k^2} \leqslant \frac{\pi^2}{6} - x$.

Exercice 5. À l'aide d'une boucle `while`, écrire la fonction `eucl (m, n)` qui calcule le pgcd de deux entiers m et n à l'aide de l'algorithme d'Euclide. (On rappelle qu'il s'agit du dernier reste non nul dans la suite des divisions euclidiennes.)

Exercice 6. 1. Déterminer ce que fait la fonction suivante.

```
def abso (x):
    if x >= 0:
        return x
    else:
        return -x
```

2. Écrire la fonction `testliste (L)` qui pour une liste `L` affiche les éléments qui sont pairs. (La taille d'une liste `L` s'obtient avec `len (L)`.)
3. Créer `L1` la liste des restes modulo 7 des carrés des entiers entre 0 et 7 et tester `testliste` sur `L1`.

Exercice 7. 1. À quoi servent les commandes `ord('a')`, `ord('b')`, `ord('z')`, `chr(97)`, `chr(98)`, `chr(122)` ?

2. Ecrire une fonction qui prend en entrée une chaîne de caractères constituée uniquement de lettres minuscules et qui renvoie une liste de nombres. Puis faire en sorte que les nombres soient compris entre 1 et 26.
3. Écrire une fonction qui fait l'inverse : c'est-à-dire, qui prend une liste de nombres entiers entre 1 et 26, et qui renvoie une chaîne de lettres minuscules.
4. Refaire les deux questions ci-dessus en intégrant la possibilité de mettre des espaces dans la chaine de caractères.

(5. Envisager des applications pour transmettre des messages codés.)

Exercice 8. Dans cet exercice, f désigne une fonction numérique dérivable.

1. Écrire une fonction qui prend en arguments f et un réel x_0 et qui renvoie l'équation de la tangente à la courbe de f au point d'abscisse x_0.
2. Vérifier le résultat précédent sur quelques exemples (différents de ceux du voisin) à l'aide de `plot`.
3. Écrire une fonction `calcul_intersection` qui prend en entrée f et un réel x_0 et qui renvoie l'abscisse du point d'intersection de la tangente à la courbe de f au point d'abscisse x_0 avec l'axe des abscisses.
4. Dans cette question, on considère la fonction f définie par $f(x) = x^2 - 2$. Calculer `x1 = calcul_intersection(f, 2)`, `x2 = calcul_intersection(f, x1)`, etc. Que remarque-t-on ? Représenter graphiquement ces calculs.
5. Écrire une fonction `Newton` qui prend en arguments f, un réel x_0, un entier n et qui renvoie la liste `[x1, x2,...,xn]`, où `x1 = calcul_intersection(f, x0)`, `x2` est égal à `calcul_intersection(f, x1)`, etc.
6. Vérifier la cohérence des résultats en calculant `Newton(x^2-2,2,n)` pour des petites valeurs de n. À quoi correspond `Newton(x^2-2,-2,n)` ? `Newton(x^2+2,n)` ?
7. Trouver des rationnels très proches de $\sqrt{2}$, $\sqrt{3}$, $\sqrt[3]{2}$, du nombre d'or (défini comme l'unique solution positive de l'équation $x^2 = x + 1$).
8. Tester `Newton` avec $x^5 - \frac{13}{4}x^3 + \frac{9}{4}x$ en choisissant différents points de départ. Expliquer précisément ce qu'il se passe (on pourra tracer le graphe de $x \mapsto x^5 - \frac{13}{4}x^3 + \frac{9}{4}x$ sur un intervalle assez grand, essayer avec d'autres fonctions, ...).