TP n° 5 : examen pratique

Vous pouvez traiter les exercices dans l'ordre que vous voulez, ils ne sont pas ordonnés par ordre de difficulté (sauf le dernier, qui est sans question intermédiaire).
En fin d'épreuve, vous devrez avoir créé un fichier de code Python qui s'appellera

`votrenomdefamille.py`

où votre nom de famille sera écrit en lettres minuscules, sans accents, sans cédille, sans espaces (pour les homonymes Bonnet et Dugast, vous ajouterez en fin de nom l'initiale de votre prénom : `bonnetf.py`,...). Une bonne idée est de le créer tout de suite et de régulièrement sauvegarder votre travail. Si ça plante, vous récupérerez quand même ce qui a précédé.
Puis vous vous identifierez sur prepabellevue.org, vous afficherez l'article 1622 intitulé « Epreuve pratique d'informatique »(`http://prepabellevue.org/index.php?article=1622`) de la rubrique MPSI / Informatique et vous enverrez votre fichier en utilisant le formulaire en ligne.
En cas de problème d'envoi, signalez-le tout de suite ! Et n'attendez pas la dernière minute pour le faire !

On rappelle les fonctions Python suivantes :
- `str(x)` transforme un objet `x` en une chaîne de caractères
- `int(x)` transforme un objet `x` en un entier si cela a un sens

Sauf indication contraire, les symboles `n,k,p` désignent toujours des entiers, `L,M` des listes de nombres.

I Des équations diophantiennes

- Écrivez une fonction `racines1(n)` qui donne la liste des entiers x compris entre 0 et $n-1$ tels que $x^2 \equiv 1$ $[n]$.

  Par exemple, `racines1(8)= [1, 3, 5, 7]`.
- Écrivez une fonction `nbr1(p)` qui donne la liste contenant le nombre de solutions comprises entre 0 et $k-1$ de l'équation $x^2 \equiv 1$ $[k]$, pour k variant de 2 à p.

  Par exemple, `nbr1(8)= [1, 2, 2, 2, 2, 2, 4]` car modulo 2, il y a un seul nombre tel que $x^2 \equiv 1$ $[2]$ et modulo 8, il y en a 4.

II Encore des équations diophantiennes

- Écrivez une fonction `puis(a,n)` qui calcule la liste des nombres $a^0, a^1, a^2, a^3, \dots$ qui sont inférieurs ou égaux à n.

  Par exemple, `puis(2,10)= [1,2,4,8]`.
- Écrivez une fonctions `sommes(L,M)` qui calcule la liste de toutes les sommes possibles entre deux éléments l'un pris dans la liste `L`, l'autre pris dans `M`.

  Par exemple, `sommes([1,3,5], [4,8,9])= [5, 9, 10, 7, 11, 12, 9, 13, 14]`.

— Écrivez une fonction `sol(n,a,b)` qui indique si un entier n peut s'écrire sous la forme $a^k + b^\ell$ où $(k,\ell) \in \mathbb{N}^2$.

  Par exemple, `sol(17,2,3)= True` car $17 = 2^3 + 3^2$, `sol(12,2,3)= False`.

— Écrivez une fonction `sols(n,a,b)` qui donne la liste des couples (k,ℓ) tels que $n = a^k + b^\ell$, rangés dans l'ordre croissant des k (la liste est vide s'il n'y a pas de solution).

  Par exemple, `sols(17,2,3)= [(3,2), (4,0)]` car $17 = 2^3 + 3^2 = 2^4 + 3^0$, `sol(12,2,3) = []`.

III Algorithme d'Euclide

— Écrivez une fonction `grandnombre(n)` de paramètre $n \geqslant 1$ qui renvoie le nombre dont l'écriture en base 10 est constituée des écritures en base 10 des entiers 1,2,3,...,n collées les unes derrière les autres.

  Par exemple, `grandnombre(4)= 1234`, `grandnombre(12)= 123456789101112`.

— Écrivez une fonction `pgcd(a,b)` qui calcule le pgcd des deux entiers positifs a et b.

  Par exemple, `pgcd(12,30)= 6`, `pgcd(grandnombre(11),grandnombre(19))= 13`.

— Écrivez une fonction `pgcdmax(i,n)` qui calcule le couple (p,m) où m est le plus grand pgcd des couples `grandnombre(k), grandnombre(k+i))` pour k variant de 1 à n, et p la valeur de k correspondante.

  Par exemple, `pgcdmax(2,100)= (96,138)`

IV Listes convergentes

— Écrivez une fonction `valeurs(f,n)` qui calcule la liste des valeurs `[f(0), f(1), ..., f(n-1)]` d'une fonction `f` définie sur $\mathbb{R}_+$.

  Par exemple, `valeurs(sin, 5)= [0.0, 0.841..., 0.909..., 0.141..., -0.756...]` (tous les chiffres n'ont pas été écrits sur le sujet pour gagner de la place).

— On appelle limite d'une liste de nombre la moyenne entre le plus grand et le plus petit des 10% derniers termes. Écrivez une fonction `limite(X)` de paramètre une liste de nombres `X` qui renvoie la valeur de sa limite.

  Par exemple, si `X` est la liste des valeurs de la fonction $x \mapsto 10\frac{\sin x}{x+1}$ prises aux points entiers $0, 1, \ldots, 19$, alors `limite(X)= -0.16015...`

— Soit $\varepsilon > 0$. Une liste de nombres de limite ℓ est dite convergente à ε près si les 10% derniers termes de la liste sont tous dans l'intervalle $[\ell - \varepsilon, \ell + \varepsilon]$. Écrivez une fonction `convergente(X,e)` de paramètres une liste de nombres `X` et un nombre strictement positif `e` qui indique si la liste `X` est convergente à `e` près.

  Par exemple, si `X` est la liste précédente, alors
  `convergente(X,0.3)= True` et `convergente(X,0.2)= False`

V Plus proche voisin

Dans cet exercice, on appelle point une liste de deux nombres (qui sont des coordonnées d'un vrai point géomtrique), les lettres `p,q` désignent des points et `P` une liste de points.

— Écrivez une fonction `dist(p,q)` qui calcule la distance entre deux points.

  Par exemple, `dist([1,2],[2,0])= 2.23...`

— Écrivez une fonction `ppv(p,P)` qui donne le plus proche voisin de `p` dans la liste `P`, c'est-à-dire le point de la liste `P` qui est à distance minimale de `p` (s'il y en a plusieurs, on en donne un qui convient).

  Par exemple, si `P = [[0,0],[1,1],[2,0],[3,2],[4,0]]`, alors `ppv([2.5,-0.5],P)= [2,0]`

— Écrivez une fonction `ppv2(p,P)` qui donne une liste contenant les deux plus proches voisins de `p` dans la liste `P` (s'il y a plusieurs solutions, on en donne une).

  Par exemple, si `P` est la liste précédente, alors `ppv2([2.5,-0.5],P)= [[2,0],[4,0]]`

— Écrivez une fonction `ppvs(p,P,k)` qui donne une liste contenant les `k` plus proches voisins de `p` dans la liste `P` (s'il y a plusieurs solutions, on en donne une). Bien sûr on supposera que `k` est plus petit que la longueur de la liste.

VI Plus longue sous-suite croissante

On se donne `L` une liste de nombres. Écrivez une fonction `plssc(L)`, qui détermine la plus longue sous-suite croissante de `L` (une sous-suite est une liste constituée d'éléments de `L`, dans le même ordre que dans `L`, mais pas forcément consécutifs). S'il y a plusieurs solutions, une d'entre elle est renvoyée.
Par exemple, `plssc([-6,4,-4,3,5,8,9,6])= [-6,-4,3,5,8,9]`.