DS 1 - Informatique - 1h

Exercice 1
On considère des listes dont les éléments sont eux-même des listes représentant des points du plan : le nom du point, son abscisse, son ordonnée. Par exemple :

```
L=[['A',-1,1],['B',2,3],['E',-3,-2],['F',2,5],['M',7,-5]]
```

1. Écrire une fonction `init(L)` qui prend en argument une liste de points et qui renvoie un dictionnaire dont les clés sont les noms des points et dont les valeurs sont les coordonnées correspondantes.
2. Écrire une fonction `distance(dico,point1,point2)` qui prend en entrée :
   — un dictionnaire contenant des points et leurs coordonnées (comme celui renvoyé par la fonction `init`),
   — deux chaînes de caractères qui sont des noms de points du dictionnaire,

   et qui renvoie la distance entre les points précisés en paramètre.
3. Écrire une fonction `distmin(dico)` qui renvoie les deux points les plus proches (`dico` étant un dictionnaire comme celui renvoyé par la fonction `init`).

Exercice 2
On considère des clés sur un ensemble des 26 caractères : `a,b,... , z`. On associe à chaque lettre son rang dans l'ordre alphabétique : 1 pour `a` jusqu'à 26 pour `z`.
Pour chaque clé, on associé l'entier $a_0 + a_1 \times 26 + a_2 \times 26^2 + \dots a_{n-1} \times 26^{n-1}$ où $a_0, \dots, a_n$ sont les rangs des caractères de la clé.
Par exemple si la clé est le mot `python` alors $a_0 = 16$, $a_1 = 25$, $a_2 = 20$, $a_3 = 8$, $a_4 = 15$, $a_5 = 14$ et l'entier associé est :

$$x = 16 + 25 \times 26 + 20 \times 26^2 + 8 \times 26^3 + 15 \times 26^4 + 14 \times 26^5 = 173348698$$

On rappelle que si a et b sont deux entiers positifs ($b \neq 0$), le reste de la division euclidienne de a par b s'obtient grâce à la commande `a%b`.

On pourra utiliser la fonction `ord` qui renvoie le code ASCII du caractère donné en paramètre :

```
>>> ord('a')
97

>>> ord('b')
98
```

1. Écrire une fonction python `hachage(cle,N)` qui prend en entrée une chaîne de caractères (qui représente une clé) et un entier N, qui calcule x l'entier qui lui est associé, qui renvoie le reste de la division euclidienne de x par N.
   Que renvoie la commande `hachage('python',100)` ?

   Dans la suite, la valeur de hachage d'une clé est la valeur renvoyée par la fonction `hachage`.
2. Écrire une fonction `collisions(listecle,N)` qui prend en entrée une liste de clés et un entier `N` et qui renvoie le nombre de collisions, c'est-à-dire le nombre de clés de la liste ayant même valeur de hachage qu'une autre clé de la liste.
3. Écrire une fonction `enregistrement(listecle,N)` qui prend prend en entrée une liste de clés et un entier `N` et qui renvoie une liste `L` de longueur `N` telle que, pour chaque clé de la liste de clés, `L[i]` contient la clé ayant pour valeur de hachage `i`.
   S'il y a une collision, la fonction renvoie une liste vide.
4. Écrire une fonction `verification(L,i)` qui vérifie que `L[i]` contient une clé ayant pour valeur de hachage `i` (`L` est une liste de clés et `i` est un entier compris entre 0 entre $N-1$ où N est la longueur de `L`).

Exercice 3 (Les k plus proches voisins)
Nous avons dans le tableau ci-dessous, la taille (en cm), le poids (en kg) et la taille du T-Shirt (L ou M) acheté par des clientes d'un magasin de vêtements.

N	Height (in cm)	Weight (in kg)	T Shirt Size
0	158	58	M
1	158	59	M
2	158	63	M
3	160	59	M
4	160	60	M
5	163	60	M
6	163	61	M
7	160	64	L
8	163	64	L
9	165	61	L
10	165	62	L
11	165	65	L
12	168	62	L
13	168	63	L
14	168	66	L
15	170	63	L
16	170	64	L
17	170	68	L

Les données sont stockées (dans cet ordre) dans une liste `Donnees`. Chaque élément de cette liste est une liste `[[h,w],s]` où les variables `w` et `h` sont des entiers et `s` est la lettre `'M'` ou `'L'`.

Par exemple `Donnees[0]` est la liste `[[158,58], 'M']`

On rappelle que pour tous points A et B de coordonnées respectives (x_A, y_A) et (x_B, y_B), la distance entre A et B est $\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$.

1. Qu'affichent les instructions :
   (a) `print(Donnees[1][0])`
   (b) `print(Donnees[0][1])`
   (c) `print(Donnees[1][0][1])`
2. Écrire un script permettant d'afficher les données : taille (en cm) en abscisses, poids (en kg) en ordonnées, une croix bleue pour la taille M, une croix rouge pour la taille L.
   On utilisera la fonction `plot` du module `matplotlib.pyplot`.
3. Écrire une fonction `distance(A,B)` qui retourne la distance entre deux points du plan. Les variables `A` et `B` données en argument sont des listes de deux réels.
4. Recopier et compléter la fonction `distancevoisins` de paramètre `X` (une liste de deux entiers), qui calcule la distance entre `X` et chaque cliente (représenté par sa taille et son poids) de la liste et qui retourne la liste des couples (numéro de client, distance).

```
def distancevoisins(X):
    ----------------------------------------
    for i in    ---------------------------
        d=      ---------------------------
        L.append([i,d])
    ----------------------------------------
```

5. L'instruction `L.sort(key=lambda l:l[i])` classe les éléments de la liste `L` par ordre croissant suivant sa colonne numéro `i`.

   Écrire une fonction `kppv` de paramètre `X` (une liste de deux entiers) qui retourne la liste `L5` des numéros des cinq clientes les plus proches de `X`.
6. Monica, une nouvelle cliente, veut acheter un T-shirt. Elle mesure 1,61m et pèse 61kg. Écrire un script qui donne la taille du T-shirt que devrait acheter Monica en utilisant l'algorithme des k plus proches voisins (avec $k = 5$).