

Avec une écriture **claire** et **lisible**, veuillez répondre en **numérotant** convenablement chaque réponse.

Exercice 1 :

1. Générer au hasard une liste de points à coordonnées entières :

- Abscisses comprises entre x_{\min} et x_{\max} ,
- Ordonnées comprises entre y_{\min} et y_{\max}

(où x_{\min} , x_{\max} , y_{\min} , y_{\max} sont des entiers, variables globales).

Le nombre de points de cette liste sera généré au hasard entre nb_{\min} et nb_{\max} (paramètres de la fonction). Vous utiliserez pour cela la fonction `randint` du module `random`.

```
from random import randint
xmin, xmax = -20, 20
ymin, ymax = -20, 20
def genereListePoints(nbmin, nbmax):
    """
    nbmin -- entier
    nbmax -- entier
    précondition: nbmin < nbmax
    Renvoie une liste de couples d'entiers,
    le premier élément de chaque couple
    est compris entre xmin et xmax (au sens large),
    le second élément de chaque couple est
    compris entre ymin et ymax (au sens large).
    Les couples (x, y) sont générés au hasard.
    Le nombre de couples est généré au hasard, entre nbmin et nbmax.
    """
    pass
```

2. Écrire une fonction prenant en paramètres 4 nombres x , y , a , b et renvoyant la distance euclidienne entre les points $M(x,y)$ et $A(a,b)$.

$$AM = \sqrt{(x_M - x_A)^2 + (y_M - y_A)^2}$$

```
from math import sqrt
def distance(x,y,a,b):
    """
    Renvoie la distance euclidienne entre A(a,b) et M(x,y)
    """
    pass
```

3. Écrire une fonction python prenant en paramètres :

- une liste de points telle que celle de la question 1 ci-dessus.
- un nouveau point A de coordonnées entières (abscisse entre x_{\min} et x_{\max} et ordonnée entre y_{\min} et y_{\max}).

et renvoyant le point de la liste qui est le plus proche de A.

Indication : on dispose de la fonction prédéfinie **min()**. Cette fonction accepte deux paramètres. Une liste où le minimum sera calculé, et un paramètre facultatif **key** où on peut passer une fonction définissant le critère de mesure utiliser pour déterminer l'élément minimum de la liste.

```
def plusProcheVoisin(listePoints, x, y):
    """
    listePoints -- liste de points à coordonnées entières
    x -- entier entre xmin et xmax
    y -- entier entre ymin et ymax

    Renvoie le point de la liste listePoints le plus proche de (x,y)
    """
    pass
```

Exercice 2 :

On veut calculer les fréquences d'apparition des lettres dans la langue française. Nous choisissons des fragments de texte depuis des articles quelconques. Soit l'exemple suivant :

S = "Paul Pogba va-t-il être appelé par Didier Deschamps pour la prochaine Coupe du monde au Qatar (20 novembre - 18 décembre) ? Le milieu de terrain de la Juventus, blessé au genou droit depuis le début de la saison, ne devrait pas retrouver la compétition avant le 10 novembre et un match de Serie A à l'Hellas Vérone. Son entraîneur Massimiliano Allegri a assuré vendredi en conférence de presse que Pogba manquerait les trois prochains matches de son équipe."

1. En utilisant les boucles, créer un dictionnaire qui regroupe le nombre d'occurrences de chaque lettre qui apparait dans la chaine S. une lettre accentuée compte pour une lettre à part entière. Tout caractère autre qu'une lettre n'est pas intéressant pour cette étude.
2. Même question, mais cette fois en utilisant l'écriture/création en compréhension.

Exercice 3 :

1. Vos notes du deuxième semestre de la précédente année universitaire sont regroupées dans un tuple par étudiant selon l'ordre suivant :
(mathématiques, physique, chimie, algorithmique, électronique, français, anglais, économie)

Une liste **etudiants** contient les notes de tous les étudiants de votre classe :

etudiants = [('nom_0 prenom_0', (math_0, phy_0, ..., eco_0)), ...]

Écrire une seule instruction pour trier la liste **etudiants** dans l'ordre décroissant de la moyenne des notes obtenues dans toutes les matières.

2. Donner la dernière note de la classe obtenue en économie ainsi que le nom de l'élève qui l'a obtenue.