

Exercice 1 :

Soit A un entier naturel. Écrire un programme qui permet de savoir si l'année A est bissextile ou non. On rappelle qu'une année est bissextile si elle est soit divisible par 4 mais non divisible par 100, soit divisible par 400.

Exercice 2 :

Soit L une liste de réels. Écrire un programme qui calcule la somme des éléments de L. Faire de même avec le produit.

Exercice 3 :

définir la liste : liste = [17, 38, 10, 25, 72], puis effectuez les actions suivantes :

- triez et affichez la liste ;
- ajoutez l'élément 12 à la liste et affichez la liste ;
- renversez et affichez la liste ;
- affichez l'indice de l'élément 17 ;
- enlevez l'élément 38 et affichez la liste ;
- affichez la sous-liste du 2^e au 3^e élément ;
- affichez la sous-liste du début au 2^e élément ;
- affichez la sous-liste du 3^e élément à la fin de la liste ;
- affichez la sous-liste complète de la liste ;
- affichez le dernier élément en utilisant un indexage négatif.

Bien remarquer que certaines méthodes de liste ne retournent rien.

Exercice 4 :

On veut réaliser une fonction occurrences(chaine) qui compte le nombre d'occurrences de chaque caractère d'une chaîne donnée, c'est à dire qu'elle compte le nombre de fois qu'apparaît chaque lettre, chiffre, espace, etc... de la chaîne.

Le résultat sera sous la forme d'un dictionnaire où chaque clé sera un caractère de la chaîne et la valeur associée sera le nombre d'occurrences de ce caractère.

Exemple : occurrences('tortue') renvoie {'t':2, 'o':1, 'r':1, 'u':1, 'e':1}

Exercice 5 :

Écrire un module de calcul des racines du trinôme réel : $ax^2 + bx + c$.

Le module définit une fonction trinome avec les trois paramètres du trinôme, a, b et

c. La fonction doit retourner un tuple dont le premier élément est le nombre de racines du trinôme (0, 1 ou 2), et les autres éléments sont les racines éventuelles.

Testez votre fonction avec les trois jeux de valeurs suivantes : 1,-3, 2, 1,-2, 1 et 1, 1, 1.

Exercice 6 :

Définir une classe MaClasse possédant les attributs suivants :

- données : deux attributs de classes : $x = 23$ et $y = x + 5$.
- méthode : une méthode affiche contenant un attribut d'instance $z = 42$ et les affichages de y et de z .

Dans le programme principal, instanciez un objet de la classe MaClasse et invoquez la méthode affiche.

Exercice 7 :

Définir une classe Rectangle avec un constructeur donnant des valeurs (longueur et largeur) par défaut et un attribut `nom = "rectangle"`, une méthode d'affichage et une méthode surface renvoyant la surface d'une instance.

Définir une classe Carre héritant de Rectangle et qui surcharge l'attribut d'instance : `nom = "carré"`.

Dans le programme principal, instanciez un Rectangle et un Carre et affichez-les.