# Fonctions & Listes

Exercice 1 : **Manipulation de fonctions**

## Partie 1 : Fonction mathématique

Ecrire une fonction qui, étant donné un réel $x$, retourne l'image de $x$ par la fonction $f$ de $\mathbb{R}$ dans $\mathbb{R}$ telle que :

$$\forall x \in \mathbb{R}, f(x) = x^4$$

Vérifier cette fonction pour les valeurs 2, 3 et 4.

## Partie 2 :

Écrire une fonction qui affiche l'aire d'un cercle.

Vérifier cette fonction avec le cercle trigonométrique.

## Partie 3 :

Ecrire une fonction qui, étant donné deux entiers $n$ et $m$, renvoie `True` si les deux entiers ont la même parité, et `False` sinon.

Exercice 2 : **Manipulation de listes**

## Partie 1 : Création et manipulation de liste

Une liste est définie entre crochets, avec un séparateur entre les éléments : la virgule.

```
maliste = [’Lun’, ’mar’, ’Sam’, ’ Dim’]
```

Il est aussi possible de former une liste par concaténation à partir de plusieurs listes en utilisant le symbole + entre ces listes.

```
une_autre_liste = [0,1]+[2,3]
```

Pour extraire un élément d'une liste, on utilise son index (numéro d'apparition dans la liste), qui commence à 0 pour une lecture de gauche à droite, et à -1 pour une lecture de droite à gauche.

```
maliste[2]
maliste[-3]
```

Pour extraire une sous-liste d'une liste, il est nécessaire d'indiquer l'index du premier élément et l'index du dernier élément plus un, séparés par deux points.

```
maliste[0:2]
```

Voici quelques fonctions de bases sur les listes.

| | |
|---|---|
| `range(start, stop, pas) -> list` | Renvoie une liste de nombre, commençant à start, finissant à stop, deux valeurs successives étant espacées du pas |
| `len(list) -> int` | Renvoie le nombre d'éléments de la liste. |
| `value in list` | Renvoie un booléen indiquant si l'élément est présent dans la liste. |
| `max(list) -> value`<br>`min(list) -> value` | Si les éléments sont comparables : renvoie le plus grand ou le plus petit élément. |
| `sorted(list) -> list`<br>`sorted(list,reverse=True) -> list` | Si les éléments sont comparables : renvoie une liste des éléments triés par ordre croissant, ou décroissant. |

**Q1 :** Calculer la longueur de la liste `maliste`.
**Q2 :** Ajouter `'jeu'` à la fin de `maliste`.
**Q3 :** Afficher le deuxième élément, l'avant-dernier élément.
**Q4 :** Trier cette liste donnée sous forme alphabétique croissante puis décroissante.
**Q5 :** Dans un programme, à l'aide d'une boucle, construire une nouvelle liste `newliste`, de même contenu que `maliste`, mais d'ordre inversée. (attention, il est parfois nécessaire de définir une nouvelle variable de liste avant de l'utiliser. Par exemple : `newliste=[]`)

Exercice 3 : **Recherche d'intrus**

On considère la liste hétérogène ['lun','mar',3,'jeu','ven','sam','dim']

**Q1 :** Proposer un programme qui, à l'aide d'une boucle, recherche l'intrus (élément de type différent) dans une liste sachant que l'intrus est unique et qu'il n'est pas en première position. Faire afficher l'index de l'intrus
**Q2 :** Compléter le programme pour qu'il affiche la liste dont il aura retiré l'intrus.

Exercice 4 : **l'algorithme de Luhn (exercice d'entrainement)**

Certains numéros administratifs (carte bancaire, numéro d'identification SIREN d'une entreprise, plaque d'immatriculation dans certains pays, IMEI d'un téléphone, carte SNCF) doivent respecter certaines règles contraignantes qui limitent les contrefaçons. L'algorithme de Luhn en est une. Il procède en trois étapes.

- L'algorithme remplace un chiffre sur deux, en commençant par l'avant dernier et en se déplaçant de droite à gauche. Il est remplacé par son double. Si un chiffre qui est multiplié par deux est plus grand que neuf, on y retranche neuf, pour qu'il soit compris entre 0 et 9.
- La somme de tous les chiffres obtenus est effectuée.
- Si le reste de la division par 10 est égal à zéro, alors le code original est valide.

**Q1 :** Proposer une fonction qui, à une liste de chiffre correspondant au code, renvoie vrai si l'algorithme de Luhn vérifie le code.

**Q2 :** Vérifier avec le numéro d'identification 972-487-086, ou avec votre numéro de carte bleue.

Exercice 5 : **Moyenne et variance**

Le fichier `note.txt` contient les notes obtenues par les élèves de PCSI à un DS.
Pour charger le fichier sous forme d'une liste, utiliser le programme `lecture.py`
Une liste L a été crée :

```
print(L)
```

Il est possible d'afficher un graphe des notes (éléments de la liste) en fonction de l'index dans la liste :

```
plot(L)
```

La courbe représentée n'a aucune signification : les grandeurs tracées ne sont pas continues. On préfèrera :

```
plot(L,'o')
```

Il est aussi possible de tracer l'histogramme associé à la répartition de ces notes :

```
hist(L,range(20))
```

Ou, pour faire encore plus joli, on peut utiliser la fonction fournie par le professeur : `affichage_note.py`

On cherche à extraire les données statistiques de cette liste.

En utilisant un programme utilisant une ou des boucles for :

Q1 : Déterminer la moyenne des notes
Q2 : Déterminer la variance

Exercice 6 : **Tracé de fonction**

Créer la liste X=[0,0.01,0.02,...,2]
Créer la fonction :

$$f(x) = \frac{x}{1+x^3}$$

Créer la liste Y=[f(0),f(0.01),f(0.02),...,f(2)]

Au début de votre programme, taper :

```
import matplotlib.pyplot as plt
```

puis en fin de programme, taper :

```
plt.plot(X,Y)
```