TD 2 -
Prise en main

Informatique
MPSI-PCSI - Lycée Thiers

Exercice 1.

Exercice 1. Ouvrir le logiciel Pyzo. Dans la console, obtenir les valeurs (approchées) de :

- 2^8, 2^{10}, 2^{16},
- $\sqrt{3}$, $\sqrt[4]{5}$, $\sqrt[3]{5}$,
- le quotient dans la division euclidienne de 11^{10} par 9^{12} ; de ces 2 nombres lequel est le plus grand ?
- le reste dans la division euclidienne de 3^7 par 5.
- Quelle expression permet d'obtenir le chiffre des 100 millions de 2^{100} ?

Exercice 1. Corrigé

On saisit les expressions au prompt dans la console ;
à l'exécutiion (Entrée), on obtient le résultat :

```
In [1]: 2**8 , 2**10 , 2**16
Out[1]: (256, 1024, 65536)
In [2]: 3**0.5 , 5**0.25 , 5**(1/3)
Out[2]: (1.732050807568772, 1.495348781221205,
1.709975946676697)
In [3]: 11**10 // 9**12      # On constate que 9**12 > 11**10
Out[3]: 0
In [4]: 3**7 % 5
Out[4]: 2
In [5]: (2**100 // 10**8) % 10
Out[5]: 7
```

Exercice 2.

Exercice 2.

- Ecrire une fonction `cosh` qui retourne la valeur du *cosinus hyperbolique* de son paramètre :

$$\forall\, x \in \mathbb{R}, \qquad \cosh(x) = \frac{\exp(x) + \exp(-x)}{2}$$

- Ecrire une fonction `sinh` qui retourne la valeur du *sinus hyperbolique* de son paramètre :

$$\forall\, x \in \mathbb{R}, \qquad \sinh(x) = \frac{\exp(x) - \exp(-x)}{2}$$

Puis obtenir dans la console les valeurs retournées par `cosh` et `sinh` en 0 et $\ln(2)$.

Exercice 2. Corrigé

```
## exo 2
from math import exp

def cosh(x):
    return (exp(x)+exp(-x))/2

def sinh(x):
    return (exp(x)-exp(-x))/2
```

Après exécution, on les appele dans la console pour les valeurs demandées :

```
In [1]: cosh(0)
Out[1]: 1
In [2]: from math import log
In [3]: cosh(log(2))
Out[3]: 1.25
In [4]: sinh(0)
Out[4]: 0
In [5]: sinh(log(2))
Out[5]: 0.75
```

Exercice 3.

Exercice 3.

1. Dans l'éditeur saisir le code des 2 fonctions suivantes, sans les commentaires, qui permettent, pour la première de tracer le graphe d'une fonction sur un intervalle, pour la seconde de fermer la fenêtre.

2. Les appeler pour tracer le graphe de $\cosh$ et $\sinh$ au dessus de l'intervalle $[-2;2]$.
3. Modifier le code des fonctions `cosh` et `sinh` de l'exercice 2 en utilisant `print` au lieu de `return`, puis retenter le tracé du (2). Que se passe-t-il ?

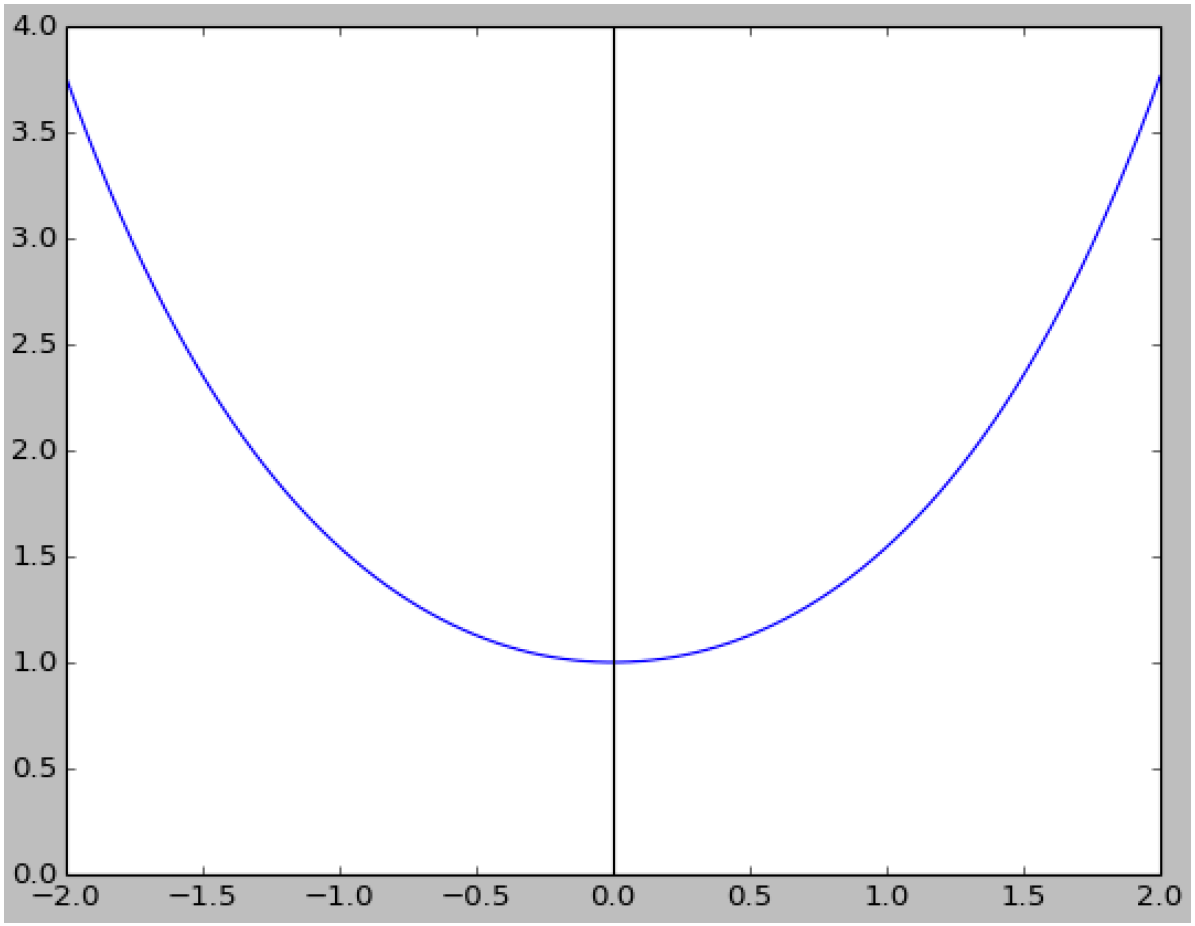
Exercice 3. Enoncé

```
from numpy import linspace    # Import de la fonction linspace
import matplotlib.pyplot as plt     # Import du module de tracé

def graphe(f,a,b):    # Graphe de f au dessus de [a,b]
    plt.figure(1)    # Ouverture de Figure 1
    X = linspace(a,b,100)    # tableau des abscisses
    Y = [f(x) for x in X]     # tableau des ordonnées
    plt.plot(X,Y)         # Tracé
    plt.axhline(color='black')    # axe des abscisses
    plt.axvline(color='black')    # axe des ordonnées
    plt.show()

def ferme():
    plt.figure(1)    # Ouverture de Figure 1
    plt.close()    # fermeture
```

Exercice 3. Corrigé

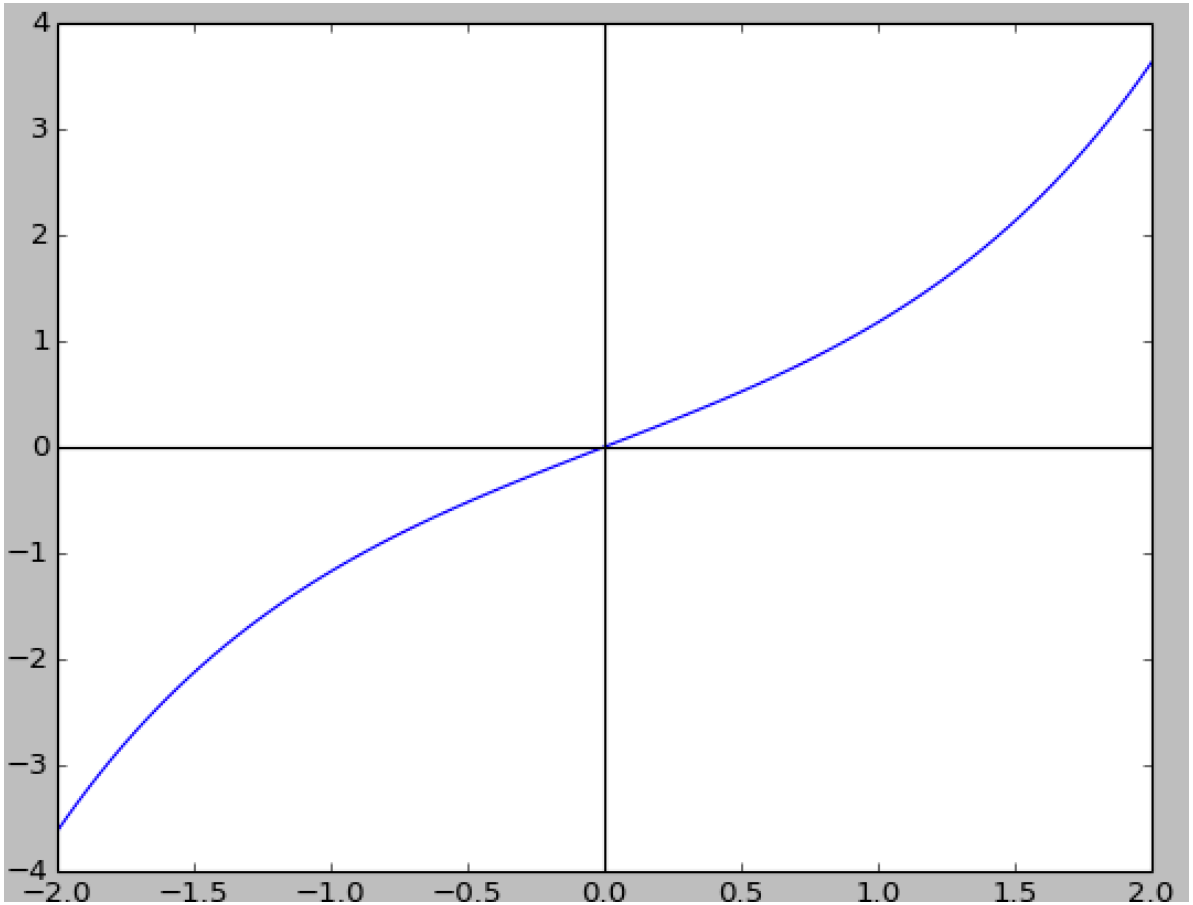
2.

```
In [1]: graphe(cosh,-2,2)
```



Exercice 3. Corrigé

```
In [1]: graphe(sinh,-2,2)
```



Exercice 3. Corrigé

3. En modifiant le code des fonctions `cosh` et `sinh` avec la fonction `print` :

```
def cosh(x):
    print((exp(x)+exp(-x))/2)

def sinh(x):
    print((exp(x)-exp(-x))/2)
```

Le tracé de `graphe(cosh,-2,2)` est vide de courbes. En effet sans utiliser `return` ces fonctions ne renvoient plus à l'appel de `graphe` les valeurs calculées.

Exercice 4.

Exercice 4. Au-dessus de l'intervalle $[0,1]$, tracer dans une même figure les graphes des fonctions :

$$x \mapsto x \quad ; \quad x \mapsto x^2 \quad ; \quad x \mapsto x^3 \quad ; \quad x \mapsto x^4 \quad ; \quad x \mapsto x^5 \quad ; \quad x \mapsto x^6$$

Exercice 4. Corrigé

On commence par définir les 6 fonctions :

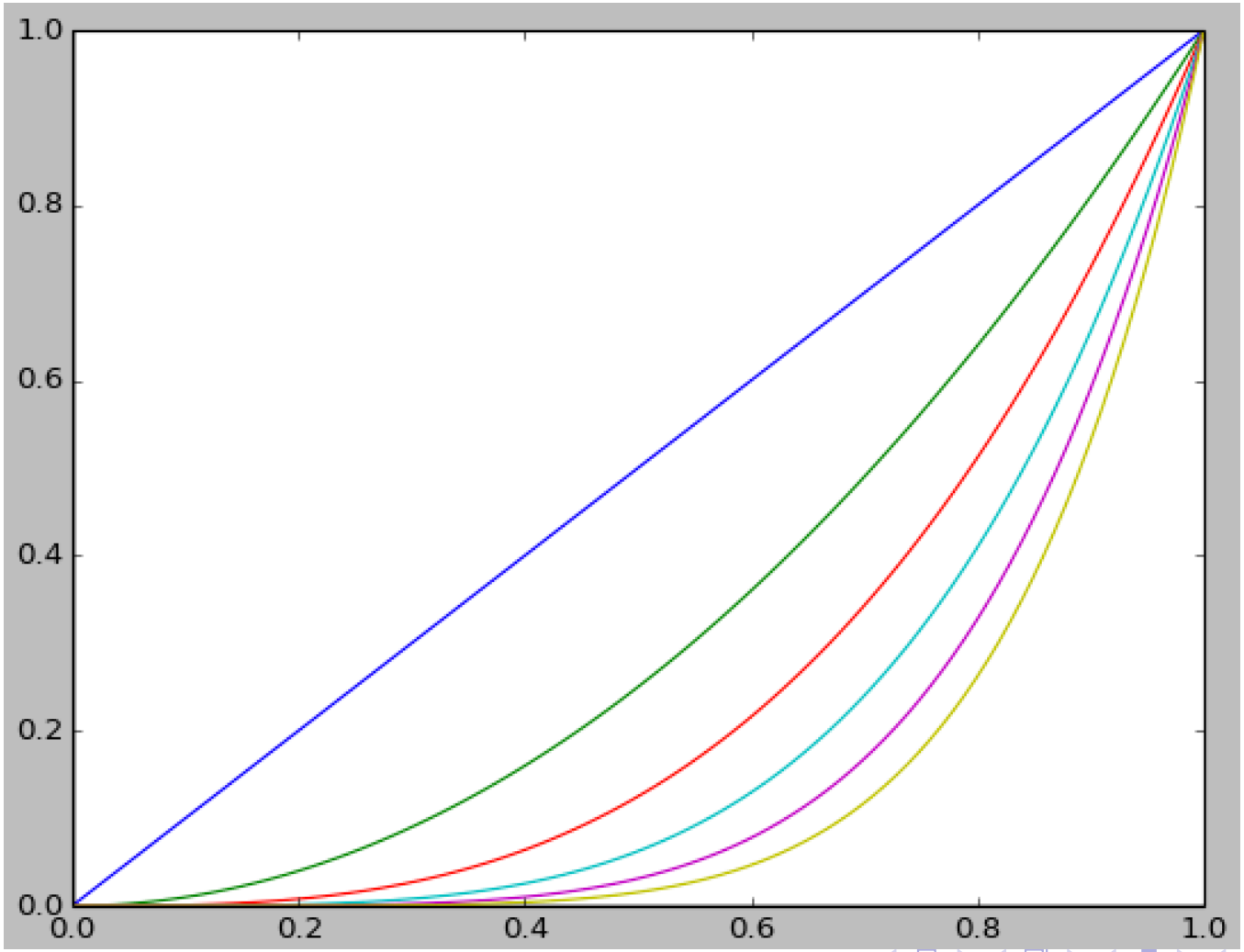
```
def f1(x):
    return x

def f2(x):
    return x**2

# etc...
def f6(x):
    return x**6
```

Puis on appelle dans la console, 6 fois la fonction `graphe` avec pour premier paramètre `f1`,`f2`,... ,`f6`, et pour derniers paramètres 0 et 1 :

```
In [1]: graphe(f1,0,1)
...
In [6]: graphe(f6,0,1)
```

Exercice 4. Corrigé

On obtient :



Exercice 4. Corrigé

Sur certains ordinateurs cette méthode ne fonctionnera pas (elle ne tracera que la première courbe. On obtient alors le tracé avec le code :

```
plt.figure(1)          # Ouverture de Figure 1
X = linspace(a,b,100)     # tableau des abscisses
Y1 = [f1(x) for x in X]      # tableau des ordonnées de f1
plt.plot(X,Y1)                 # Tracé de f1
# etc...
Y6 = [f6(x) for x in X]      # tableau des ordonnées de f6
plt.plot(X,Y6)                 # Tracé de f6
plt.show()
```