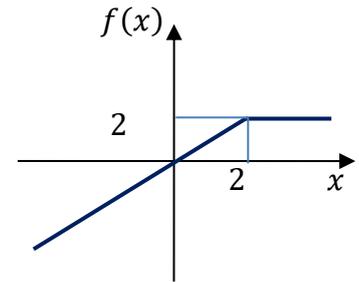


## TP5 : Fonctions, modules, fichiers

### Exercice 1 : Fonction « saturation »

Ecrire une fonction traduisant le comportement décrit par le graphe ci-contre :



### Exercice 2 : Tracé de fonction

Créer la liste  $X=[0,0.01,0.02,\dots,2]$

Créer la fonction :

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

Créer la liste  $Y=[f(0),f(0.01),f(0.02),\dots,f(2)]$

Au début de votre programme, taper :

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

puis en fin de programme, taper :

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

### Exercice 3 : Conversion d'un entier exprimé en binaire

**Q 1 :** Écrire une fonction `zero_un(s)` qui teste si une chaîne de caractères n'est formée que des caractères 0 et 1.

**Q 2 :** Écrire une fonction `conv(s)` qui renvoie l'entier représenté par une chaîne de caractères en binaire, conformément à la convention vue en cours (par exemple `conv("1011")` renvoie 11).

**Q 3 :** Écrire un programme qui demande à l'utilisateur d'entrer une chaîne de caractères puis utilise les fonctions précédentes pour afficher la valeur de l'entier représenté par cette chaîne si elle ne contient que des 0 et des 1, ou affiche une erreur sinon.

### Exercice 4 : Les années bissextiles

La fonction `isleap` du module `calendar` est décrite par son aide ci-dessous :

```
help(calendar.isleap)
```

**Q 1 :** A partir de l'aide de cette fonction, donner sa spécification.

**Q 2 :** Ecrire un programme qui affichent les années bissextiles entre 1900 et 2020.

### Exercice 5 : Gestion de calendrier

Certains logiciels de gestion de calendrier s'appuient sur la convention suivante :

- A chaque jour est codé par un entier ;
- L'origine est le 1<sup>o</sup> janvier 1900, codé par l'entier 1 ;
- Le jour suivant est codé 2, et ainsi de suite.

Proposer un programme qui affiche la date du jour correspondant à un entier indiqué par l'utilisateur. Le format à utiliser pour l'affichage de la date doit respecter l'exemple (14 février 2018) :

```
Jour : 14
```

```
Mois : 02
```

```
Année : 2019
```

Ce programme pourra utiliser les fonctions mises en place dans les exercices précédents.

**Exercice 6 : Suivi de compte bancaire**

L'objectif de cet exercice est un programme de suivi de solde bancaire. Ce programme s'appuie sur trois fonctions :

- fonction « solde » : affiche le solde du compte
- fonction « debit(somme à retirer) » : retire la somme du compte, et affiche le solde modifié
- fonction « credit(somme à ajouter) » : ajoute la somme au compte, et affiche le solde modifié

Le corps du programme est décrit ci-dessous :

```
1 Solde=1000
2 n='init'
3 while n!=0:
4     n=int(input('faite votre choix : '))
5     if n==1:
6         solde()
7     elif n==2:
8         debit(int(input('quel débit ?')))
9     elif n==3:
10        credit(int(input('quel crédit ?')))
11    elif n==0:
12        print('fin de programme')
13    else:
14        print('choix non valide')
```

**Partie 1 : Compréhension du programme**

- Q 1:** Quelles sont les variables utilisées par ce programme ?  
**Q 2:** Décrire les opérations réalisées à la ligne 8.  
**Q 3:** Quelle est l'utilité de la structure conditionnelle ici ?  
**Q 4:** Quelle est l'utilité de la structure « while » ?  
**Q 5:** Quelle est la différence entre « Solde » et « solde » ?

**Partie 2 : Finalisation**

- Q 6:** Rédiger les fonctions « solde », « credit », « debit ».  
**Q 7:** Vérifier le bon fonctionnement en exécutant le programme.