Exercices : Fonctions

Exercice n°1 : écriture d'une fonction simple

Complétez le programme **Caisse.py** suivant en ajoutant la définition de la fonction Taxe dont l'appel est fait dans la partie « test »

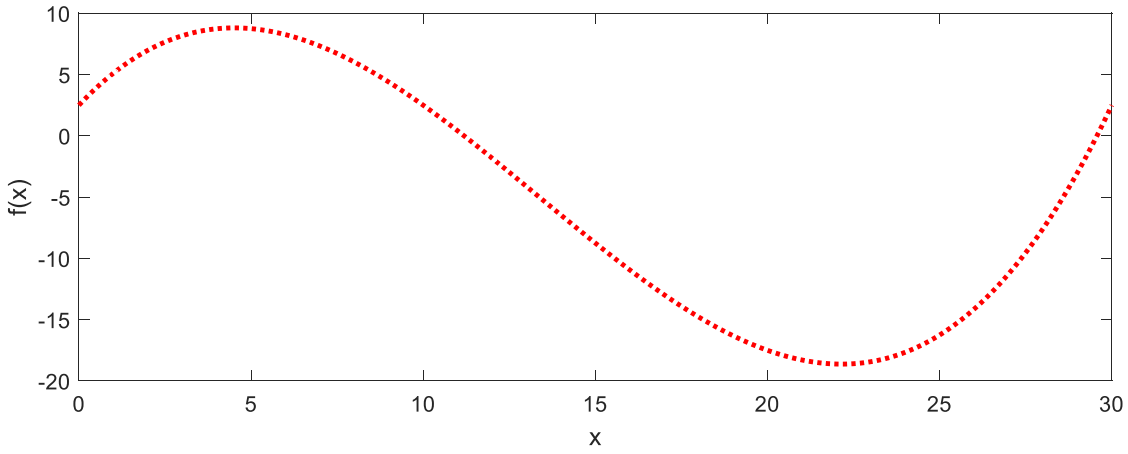
```
### Définition fonction #####
def TVA #(…à compléter)
############################

###  Test ##################

TVA = 20
PrixUnitaire = float(input("Prix Unitaire de l''article : "))
Nombre   = int(input("nombre d''article : "))
Total = Taxe(PrixUnitaire,Nombre,TVA)
print("Merci de régler la somme de ",round(Total,2))
```

Exercice n°2 : appel à une fonction simple

Soit la fonction $y = 0.01x^3 - 0.4x^2 + 3x + 2.5$ qui se correspond à la courbe suivante :



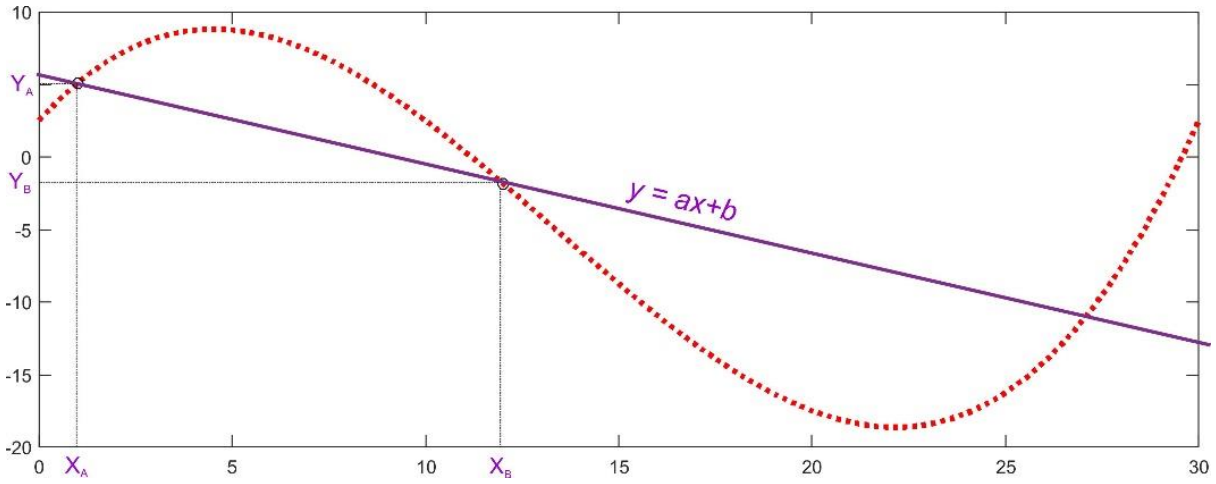
Cette fonction se code en Python comme :

```
def Effede(x) :
      fdex =  0.01x**3-0.4x**2+3*x+2.5
      return(fdex)
```

Recopiez ces trois lignes de code dans un programme python (**Fonc_Exo1.py**)

Compléter ce script pour appeler la fonction **Effede** pour la valeur $X_A = 1$. On appellera Y_A la variable contenant l'image de X_A par la fonction.

On souhaite maintenant calculer l'équation de la droite passant par (X_A, Y_A) et un point (X_B, Y_B) comme sur la figure suivante :



L'abscisse X_B sera saisi par l'utilisateur, l'ordonnée Y_B sera calculée par appel à la fonction.

Le programme se terminera par l'affichage de l'équation de la droite.

Exercice n°3 : écriture et test de fonctions

Reprendre le script `Racine_V1.py` pour le transformer en fonction. Cette fonction (`Racine_V2.py`) prendra les coefficients du polynôme en entrée et retournera les valeurs des 2 racines. On continuera d'assumer que cela ne fonctionnera pas si le discriminant est négatif.

Tester directement la fonction dans la console en donnant des valeurs arbitraires.

Exercice n°4 : écriture et test de fonctions

Créer le programme `Chronos` qui, en utilisant le module `Donne_Heure` crée les variables `H1`, `M1`, `S1` contenant le temps courant.

Utiliser ensuite la fonction `randint(min,max)` du module `random` qui permet d'obtenir une valeur entière dans un intervalle [min max] pour créer une valeur (que l'on nommera `Tempo`) comprise entre 5000 et 80 000.

Créer et appeler la fonction `AuTop` qui calcule et affiche l'heure qu'il sera `Tempo` secondes après `H1`, `M1`, `S1`.
On se rappellera qu'en Python l'opérateur // correspond à *la division entière (exemple : 452//60 donne 7)*

Exercice n°5 : Externalisation de fonctions

Soit le problème suivant :

Une baignoire de volume `Vol = 120` litres est remplie initialement de `Nb_init` litres et le débit maximum du robinet est de `Debit = 0,12` litre par seconde.

On ouvre le robinet à `Xpc` %, le débit résultant est proportionnel au taux d'ouverture du robinet.

Ecrire le module `Baignoire` contenant les deux fonctions `Deborde` et `Capa` qui permettent :

- de calculer (fonction `Deborde`) le temps que vous avez avant que le bain ne déborde.
- de calculer (fonction `Capa`) combien de litres il y aura dans la baignoire au bout de `Ts` secondes

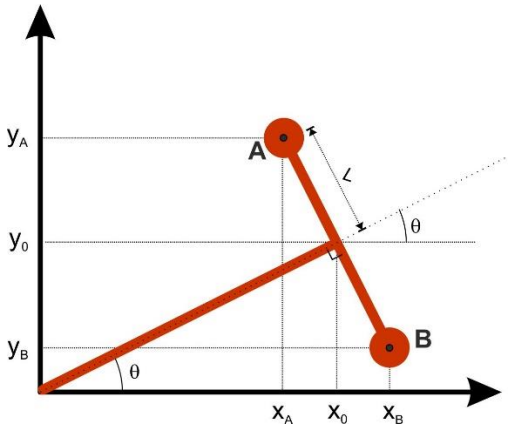
Ecrire ensuite dans un fichier différent le programme de test (`TestBaignoire.py`) de ces deux fonctions en respectant les consignes suivantes :

- `Nb_init`, `Xpc` et `Ts` seront des variables du problème. Elles seront saisies par l'utilisateur (entrées)

- **Vol** et **Debit** seront des *constantes* du programme. Elles seront codées par affectation directe dans le programme.
- On affichera le résultat des calculs faits par les fonctions **Deborde** et **Capa** .

Exercice n°5 : retour multiple

Un ingénieur mécanicien travaille sur une pièce dans le plan correspondant au schéma ci-dessous :



Il doit écrire les fonctions **CoodPendA** et **CoodPendB** qui permet de compléter le script **pendule.py** suivant :

```
x0 = input('Coordonnée en x du point de liaison :');
x0=float(x0)
y0 = input('Coordonnée en y du point de liaison :');
y0=float(y0)
L = input('Longueur du demi pendule :');
L=float(L)

xA,yA = CoodPendXA(x0,y0,L);
xB,yB = CoodPendXB(x0,y0,L);

print("Le point A est en coordonnées (",round(xA,2),", ",round(yA,2),")")
print("Le point B est en coordonnées (",round(xB,2),", ",round(yB,2),")")
```

Exercice n°5 : appels imbriqués

Ecrire les 3 fonctions Python F_1, F_2et F_3 dans même module qui correspondront aux fonctions mathématiques suivantes :

$$\begin{cases} F_1(x,\alpha) = 3x + \alpha \\ F_2(x,\alpha) = x^2 - \alpha \\ F_3(x,\alpha) = e^{(0.3x+\alpha)} \end{cases}$$

Ecrire ensuite dans le même module la fonction **y = Chouse(x,α)** qui retourne le maximum des trois fonctions pour un couple (x,α).

Testez directement dans la console la fonction **Chouse** pour différentes valeurs de **x** et de **α**.