

Fonctions

Exercice 1 : 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.

Exercice 2 : Moyenne et variance

Le fichier notes.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

- Q1:** Ecrire le programme qui crée une liste L d'entiers, correspondant aux notes contenues dans le fichier.

```
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 préférera :

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

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

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

On cherche à extraire les données statistiques de cette liste.

- Q2 :** Spécifier et écrire une fonction `moyenne` qui, à une liste d'entiers, renvoie la moyenne de ces entiers. Vérifier cette fonction sur la liste `L`.
- Q3 :** Spécifier et écrire une fonction `ecart_type` qui, à une liste d'entiers, renvoie l'écart type de ces entiers. Vérifier cette fonction sur la liste `L`.

Exercice 3 : **Manipulation de données par dichotomie**

Le fichier `liste_triee.txt` contient des valeurs numériques entières triées par ordre croissants.

- Q1 :** Ecrire un programme qui importe ces données dans la mémoire vive sous la forme d'une liste d'entiers.
- Q2 :** Ecrire une fonction qui prennent en argument une liste de données triées par ordre croissant et un nombre x , et qui retourne les deux valeurs successives de la liste encadrant strictement le nombre x (on étudiera les différents cas particuliers que l'on peut rencontrer).
- Q3 :** Ecrire une fonction qui prennent en argument une liste de données triées par ordre croissant et un nombre x , et qui insère x dans la liste au bon endroit (on étudiera les différents cas particuliers que l'on peut rencontrer). Une nouvelle liste est renvoyée.
- Q4 :** Utiliser les fonctions ci-dessus pour trier la liste de valeurs contenus dans le fichier `liste_non_triee.txt`.

Exercice 4 : **Dichotomie sur une fonction**

On cherche à déterminer les racines du polynôme $f(x)$, avec une précision ε de 10^{-16} :

$$f(x) = x^3 - 2x^2 + 1$$

Partie 1 : Prise en main :

- Créer la fonction python qui, à l'argument x , renvoie :
$$f(x) = x^3 - 2x^2 + 1$$
- Tracer la courbe associée à cette fonction dans l'intervalle $[-1; 2]$ (pas de discrétisation : 0,01).

Partie 2 : Dichotomie

- Q1 :** A quelle condition la résolution par dichotomie est-elle opérationnelle ?
- Q2 :** Rappeler les arguments nécessaires à la méthode de dichotomie
- Q3 :** A partir du tracé de la courbe de f , préciser trois intervalles $[a_i, b_i]$ ($i \in \{1,2,3\}$) incluant une racine et où la fonction est monotone.

En utilisant l'algorithme de dichotomie vue en cours, programmer une fonction qui aux trois arguments : *fonction*, $[a, b]$, ε , renvoie la racine trouvée.

- Q4 :** Quelles valeurs de racine trouvez-vous ?