TP 03

Expression, variable, fonctions et structures algorithmiques

Savoirs et compétences :

- ❒ *AA.C4 : Comprendre un algorithme et expliquer ce qu'il fait*
- ❒ *AA.C5 : Modifier un algorithme existant pour obtenir un résultat différent*
- ❒ *AA.C6 : Concevoir un algorithme répondant à un problème précisément posé*
- ❒ *AA.C8 : Écrire des instructions conditionnelles avec alternatives, éventuellement imbriquées*
- ❒ *AA.S8 : Instructions conditionnelles*
- ❒ *AA.S9 : Instructions itératives*
- ❒ *AA.S10 : Notion de fonction informatique*

Consignes

1. Commencez la séance en créant un dossier au nom du TP dans le répertoire dédié à l'informatique de votre compte.
2. Après la séance, vous devez rédiger un compte-rendu de TP et l'envoyer au format électronique à votre enseignant.
3. Ayez toujours un crayon et un papier sous la main. Quand vous réfléchissez à une question, utilisez les!
4. Essayer d'être le plus autonome possible.
5. Ce TP est à faire en binôme, vous ne rendrez donc qu'un compte-rendu pour deux. Votre fichier portera un nom du type : `tp03_durif_kleim.py`, où les noms de vos enseignants sont à remplacer par ceux des membres du binôme. Le nom de ce fichier ne devra comporter ni espace, ni accent, ni apostrophe, ni majuscule. Dans ce fichier, vous respecterez les consignes suivantes.
   - Écrivez d'abord en commentaires (ligne débutant par #), le titre du TP, les noms et prénoms des étudiants du groupe.
   - Commencez chaque question par son numéro écrit en commentaires.
   - Les questions demandant une réponse écrite seront rédigées en commentaires.
   - Les questions demandant une réponse sous forme de fonction ou de script respecteront pointilleusement les noms de variables et de fonctions demandés.

Activité 1 : Fonctions en Python

Q 1 : Écrire une fonction `moy_extr(L)` qui prend en argument une liste `L` et renvoie en sortie la moyenne du premier et du dernier élément de `L`.

Q 2 : On cherche à écrire une fonction prenant en argument une liste d'entiers et incrémentant de 1 le premier élément de cette liste.

a) Écrire une telle fonction `incr_sans_effet_de_bord`, qui ne modifie pas la liste initiale et renvoie en sortie une nouvelle liste.

b) Écrire une telle fonction `incr_avec_effet_de_bord`, qui modifie la liste initiale et ne renvoie rien en sortie (ponctuer par un `return None`).

Activité 2 : Boucles IF, FOR, WHILE

Q 3 : Indenter de deux manières différentes la suite d'instructions suivante afin que la variable `t` contienne `True` pour une indentation, puis `False` pour l'autre.

```
x = 0
y = 5
t = False
if x>=1:
```

```
t = True
if y <= 6:
t = True
```

Q 4 : Réécrire la suite d'instructions suivante de manière plus appropriée.

```
from random import randrange
# Un entier aléatoire entre 0 et 99
n = randrange(100)
if n <= 10:
    print("Trop petit")
else:
    if n >= 50:
        print("Trop grand")
    else:
        print("Juste comme il faut")
```

Q 5 : Que fait la fonction suivante? La corriger pour qu'elle coïncide avec le but annoncé.

```
def inv(n):
    """Somme des inverses des n premiers
       entiers naturels non nuls"""
    s = 0
    for k in range(n):
        x = 1/k
    s = s+x
    return s
```

Activité 3 : Algorithme glouton – Problème du rendu de monnaie

La société Sharp commercialise des caisses automatiques utilisées par exemple dans des boulangeries. Le client glisse directement les billets ou les pièces dans la machine qui se charge de rendre automatiquement la monnaie.

Objectif Afin de satisfaire les clients, on cherche à déterminer un algorithme qui va permettre de rendre le moins de monnaie possible.

La machine dispose de billets de 20€, 10€ et 5€ ainsi que des pièces de 2€, 1€, 50, 20, 10, 5, 2 et 1 centimes.

On se propose donc de concevoir un algorithme qui demande à l'utilisateur du programme la somme totale à payer ainsi que le montant donné par l'acheteur. L'algorithme doit alors afficher quels sont les billets et les pièces à rendre par le vendeur.

Q 6 : *Pour un montant d'achat donné et pour une somme donnée par le client, proposer un algorithme en pseudo code permettant de rendre le minimum de monnaie au client. Cet algorithme devra détailler la somme à rendre (nombre de pièces et nombre de billets).*

Q 7 : *Quelle structure de donnée est-il possible d'utiliser pour gérer les valeurs des billets ou des pièces ?*

Q 8 : *Quel type de variable est il préférable d'utiliser pour décompter l'argent ? Pourquoi ?*

Q 9 : *Implémenter cet algorithme dans Python.*

Q 10 : *Vérifier sur plusieurs cas que l'algorithme fonctionne.*

Activité 4 : (Facultative) Structures de boucles

Q 11 : Un taupin se lance dans un marathon d'exercices, mais se fatigue vite. Il réalise le i^e exercice en $\sqrt{i}$ minutes. Combien d'exercices arrive-t-il à faire en 4 heures ? Pour faciliter la correction, on écrira une fonction `nb_exos()` ne prenant pas d'argument et renvoyant le résultat demandé.

Activité 5 : (Facultative) Fonctions plus avancées

Q 12 : Écrire une fonction `racine(n)` prenant en argument un entier naturel `n` et renvoyant sa racine carrée comme un entier si c'est un carré parfait, comme un flottant sinon.