

TP noté 2023–2024

1. Téléchargez le fichier [tp-note-2023.py](#).
2. Ouvrez Thonny (menu Applications, Développement) et ouvrez ce fichier dans Thonny.
3. Indiquez votre nom et votre prénom dans ce fichier.
4. Répondez directement dans ce fichier. Pour toutes les questions qui demandent une rédaction, écrivez les réponses en commentaire (avec un # au début de chaque ligne).
5. À **18h**, envoyez le fichier complété par mail à olivier.baudon@labri.fr et mz@labri.fr avec comme sujet : [CHS] Rendu TP.

Exercice 1

1. Quelle est la valeur représentée en base 2 par **11011** ? Détaillez le calcul vous ayant permis d'arriver à votre réponse.
2. Inversement, comment l'entier 54 se représente-t-il en base 2 ? Expliquez le raisonnement ayant conduit à votre réponse.

Exercice 2

1. Expliquez ce que fait le programme suivant, instruction par instruction. En particulier, après chaque instruction, donnez la valeur de **x** et celle de **y**.

```
x = 3
y = 0
for i in range(x):
    y = y + 2
x = y
```

2. Plus généralement, on suppose que **x** a une valeur de départ **A** et que **y** a une valeur de départ 0. Donner la valeur finale de **x** en fonction de **A** après les instructions suivantes :

```
for i in range(x):
    y = y + 2
x = y
```

Exercice 3

Dans cet exercice, on veut écrire une fonction `somme` qui :

- affiche les entiers de 1 à jusqu'à l'entier `x` passé en paramètre de la fonction, puis
- affiche la somme des entiers pairs compris entre 1 et `x`.

Par exemple, `somme(5)` doit afficher :

```
1
2
3
4
5
Somme = 6
```

Pour tester qu'un entier `k` est pair, on utilise la valeur `k % 2 == 0`, qui vaut `True` si `k` est pair, et `False` sinon.

1. Le programme suivant contient trois erreurs **empêchant son exécution**. Expliquez en français quelles sont ces erreurs, et écrire le programme où ces erreurs sont corrigées.

```
def somme(x):
    for i in range(x)
        print(i)
        if i % 2 == 0:
            resultat = resultat + i
        return resultat
```

2. Le programme une fois corrigé ne définit pas la fonction `somme` correcte. Modifiez-le pour avoir le comportement voulu.

Exercice 4

On veut afficher des chaînes de caractères composées de plusieurs fois le même caractère. En python, si `n` est un entier et `s` est une chaîne, `n * s` est la chaîne qui consiste en `n` copies de la chaîne `s`, collées entre elles. Par exemple, `5 * 'z'` vaut `'zzzzz'`.

1. Écrire des instructions permettant d'afficher :

```
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz
```

2. Écrire une fonction `ligne(c)` permettant d'afficher une ligne de `c` caractères `'z'`, où `c` est un nombre donné en paramètre. Par exemple `ligne(4)` affichera :

```
zzzz
```

3. Modifier la fonction de la question précédente pour pouvoir afficher une ligne composée de caractères précisés dans un deuxième paramètre de la fonction : `ligne2(c, caract)` affiche une ligne faite de `c` caractères `caract`. Par exemple, `ligne2(10, '/')` affichera :

```
//////////
```

4. En utilisant soit la fonction `ligne2`, soit deux boucles `for` imbriquées, écrire une fonction `rectangle(l, c, caract)` qui affiche un rectangle de `l` lignes et `c` colonnes fait de caractères `caract`. Par exemple, `rectangle(3, 10, '/')` affichera :

```
//////////  
//////////  
//////////
```

et `rectangle(2, 5, 'a')` affichera :

```
aaaaa  
aaaaa
```

5. Écrire une suite d'instructions permettant d'afficher le motif suivant :

```
*****  
****  
***  
**  
*
```

Vous pouvez ou non utiliser la fonction `ligne2`.

6. Écrire une fonction `triangle(n)` permettant d'afficher un triangle fait de caractères `'*'` dont la première ligne a `n` caractères, la deuxième en a `n - 1`, la troisième `n - 2`, etc. et la `n`-ème ligne a `1` caractère. Par exemple, le triangle de la question précédente pourra être affiché par l'appel `triangle(5)`.

Indication. Utiliser à nouveau la fonction `ligne2`.