

TP noté 2023–2024 (rattrapage)

1. Téléchargez le fichier [tp-note-2023-r.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. Une fois le fichier complété, envoyez-le par mail à olivier.baudon@labri.fr et mz@labri.fr.

Exercice 1. Base 2

1. Quelles sont les valeurs représentées en base 2 par 111 et par 1000 ? Détaillez les calculs.
2. Répondez à la même question pour 1111 et 10000.
3. Que remarquez-vous sur les exemples des questions précédentes ? Plus généralement, quel est l'entier représenté en base 2 par $\underbrace{1 \cdots 1}_{n \text{ fois}}$? Essayez de n'utiliser que les opérateurs « puissance » et « soustraction » pour exprimer cette valeur.

Exercice 2. Diviseurs

Écrire une fonction `affiche_diviseurs` qui prend en paramètre un entier `n` et qui affiche tous les diviseurs de `n` qui sont compris entre 1 et `n`. Par exemple, `affiche_diviseurs(5)` doit afficher :

```
1
5
```

et `affiche_diviseurs(12)` doit afficher :

```
1
2
3
4
6
12
```

Pour tester qu'un entier `n` est divisible par un entier non nul `k`, vous pouvez utiliser la valeur `n % k`, qui est le reste de la division entière de `n` par `k`. En particulier, `n % k` vaut 0 uniquement si `n` est divisible par `k`. Par exemple, `12 % 6` vaut 0 (mais `12 % 5` ne vaut pas 0 car 12 n'est pas divisible par 5).

Exercice 3. Correction de programme

Dans cet exercice, on veut écrire une fonction `factorielle` qui prend un paramètre entier positif `n` et renvoie la valeur $1 \times 2 \times \dots \times n$. Par exemple, si `n` vaut 5, la fonction devra renvoyer 120 (car $1 \times 2 \times 3 \times 4 \times 5$ vaut 120).

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

```
def factorielle(n)
for k in range(1, n + 1):
    resultat = resultat * k
    return resultat

print(factorielle(5))
```

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

Exercice 4. Manipulation de texte

Écrire une fonction `garder_paires` qui prend en paramètre une chaîne de caractères `s` et telle que `garder_paires(s)` affiche la chaîne obtenue à partir de `s` en ne gardant que les caractères en position paire. Par exemple :

```
>>> garder_paires("abcde")
ace
>>> garder_paires("abcd")
ac
```

On rappelle que la première position d'une chaîne non vide `s` est 0, et sa dernière position est `len(s) - 1`.

Remarque

Par défaut, l'instruction d'affichage `print` ajoute un retour à la ligne après ce qu'elle affiche. Pour éviter cela, c'est-à-dire pour afficher un texte se trouvant dans une variable `t` sans retour à la ligne supplémentaire après `t`, vous pouvez utiliser la forme suivante de l'instruction `print` :

```
print(t, end = "")
```

au lieu de

```
print(t)
```

Par ailleurs, le caractère en position `i` de la chaîne `t` est `t[i]`. Ainsi par exemple, afficher le 1^{er} caractère de `t` sans retour à la ligne se fait par l'instruction `print(t[0], end = "")`.