

Corrigé sujet **22** - Année : 2023

Sujet 22 - 2023 ⬇

Exercice 1

```
1 def liste_puissances(a,n):
2     puissances = [a]
3     for i in range(2,n+1):
4         an = puissances[-1]*a
5         puissances.append(an)
6     return puissances
7
8 def liste_puissances_borne(a, borne):
9     if a > borne :
10        return []
11    else:
12        puissances = [a]
13        while puissances[-1]*a <= borne:
14            an = puissances[-1]*a
15            puissances.append(an)
16        return puissances
```



Commentaires

- Pour la première fonction comme on suppose $n > 0$, la liste contient au moins a , puis à chaque itération on multiplie le dernier élément du tableau (obtenu avec la notation `[-1]`) par a et on l'ajoute en fin de tableau
- Pour la seconde fonction on traite le cas de la liste vide puis on ajoute de la même façon que pour la première question sans dépasser la borne.

Exercice 2

```
1 dico = {"A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6,
2         "G": 7, "H": 8, "I": 9, "J": 10, "K": 11, "L": 12,
3         "M": 13, "N": 14, "O": 15, "P": 16, "Q": 17,
4         "R": 18, "S": 19, "T": 20, "U": 21, "V": 22,
5         "W": 23, "X": 24, "Y": 25, "Z": 26}
```

```

6
7
8 def est_parfait(mot):
9     # mot est une chaîne de caractères (en lettres majuscules)
10    code_concatene = ""
11    code_additionne = 0 # 1
12    for c in mot:
13        code_concatene = code_concatene + str(dico[c]) # 2
14        code_additionne = code_additionne + dico[c] # 3
15    code_concatene = int(code_concatene)
16    if code_concatene%code_additionne == 0 : # 4
17        mot_est_parfait = True
18    else:
19        mot_est_parfait = False
20    return code_additionne, code_concatene, mot_est_parfait

```

1. Bien comprendre que le `code_additionne` est un entier (addition des entiers), tandis que `code_concatene` est une chaîne de caractères (concaténation).
2. Les valeurs du dictionnaires sont des entiers pour concaténer avec la chaîne `code_c` il faut convertir en `str`
3. On additionne des entiers, pas de conversion nécessaire
4. Teste si le code additionné divise le code concaténé.

Attention

- Les lignes 14,15,16, et 17 se résument à `mot_est_parfait = (code_concatene%code_additionne==0)`
- Même si cela est autorisé en Python, le changement de type d'une variable tel que celui effectué ligne 13 (où `int_c` qui était une chaîne de caractères devient un entier) est largement considéré comme une mauvaise pratique de programmation.
- On aurait pu se passer du dictionnaire en utilisant les fonctions `ord` et `chr`.