

1) Création et modifications d'une liste

a) Quelle instruction permet de créer cette liste `['a', 'b', 'c', 'd', 'e', 'f']`

a-1) sans la méthode par compréhension

a-2) avec la méthode par compréhension (sachant que la fonction `chr$(n)` renvoie un caractère correspondant au n^{ième} caractère de la table ASCII : `[chr$(97)='a' ; chr$(98)='b' ...]`

b) Quelles instructions doit-on effectuer sur la liste `→ ['A', 'a', 'b', 'd', 'e']` pour obtenir au final la liste : `['a', 'b', 'c', 'd', 'e', 'f']`

c) Quelles instructions doit-on effectuer sur la liste `→ ['b', 'c', 'a', 'd', 'E', 'f']` pour obtenir au final la liste : `['a', 'b', 'c', 'd', 'e', 'f']`

2) Créer une liste par compréhension

Déclarer une liste par compréhension dont les valeurs seront `[2 × n – 2]` où n augmente de 3 en 3, et n compris dans l'intervalle `[5 ; 20]` *Résultat attendu* `[8, 14, 20, 26, 32, 38]`

3) Jeu de la devinette avec la liste des valeurs proposées.

Compléter le code ci-contre pour :

- demander un nombre à l'utilisateur tant qu'il n'a pas trouvé la valeur de la variable `[solution]`.

- mémoriser toutes les propositions de l'utilisateur dans une liste

- à la fin du jeu, afficher dans la console
→ si l'utilisateur a gagné ou perdu

→ le nombre d'essais

→ la liste des valeurs proposées par l'utilisateur pendant le jeu

```
1 from random import randint
2
3 solution = randint(0, 100)
4
5 print("###Devinette ###")
6 print("Vous avez le droit à 10 essais")
7 print("  pour trouver un nombre")
8 print("dont la valeur est comprise")
9 print("  entre 0 et 100 ...")
```

4) Fonction `sqrt ()` avec précautions

Coder une fonction qui reçoit une valeur n en argument et renvoie la racine carrée de `[n]` (fonction `sqrt()` de la bibliothèque `math`) **si et seulement si** `[n]` est une valeur positive ou nul.

Sinon, cette fonction renvoie la valeur `None`.

5) Fonction `moyenne ()`

Coder une fonction `moyenne (notes:list)→float` : qui reçoit une liste de notes comprises entre 0 et 20 et retourne la moyenne des notes de la liste.

6) Fonction `moy_coeff()` avec des coefficients

Coder une fonction `moy_coeff(note_coeff:list)→ float` : qui reçoit une liste de tuples : `[(note1, coeff1), (note2, coeff2), ...]` et qui retourne la moyenne générale de la liste de notes.

Pour information : pour accéder aux valeurs contenues dans un tuple, on utilise la même notation que pour les listes : si `t = (12, 2)` alors `val = t[0]` , `coeff = t[1]` permet de récupérer 12 dans `val` et 2 dans `coeff`.

Exemple : `[(12,2), (8,3), (14,1), (13,2), (9,8), (15,8), (12,8)]` doit retourner 11.75

7) Fonction `min_max ()`

Coder une fonction qui reçoit une liste de valeurs et renvoie un tuple (valeur minimum, valeur maximum) par rapport aux valeurs contenues dans la liste.

8) Fonction `tri ()`

Coder une fonction qui reçoit une liste de valeurs dans le désordre et renvoie cette liste triée par ordre croissant de valeurs (préciser la méthode utilisée).