

Examen de Logiciel Statistique: Python 2020-2021

Première partie.

Consignes. Ce document constitue la première partie du sujet d'examen que vous aurez à réaliser pour valider le module d'Introduction à Python. Il est à réaliser seul et plusieurs aspects seront considérés dans l'évaluation.

- **Correction du code.** Celui-ci devra réaliser ce qui est demandé.
- **Lisibilité** Les fonctions devront respecter les bonnes pratiques introduites dans la leçon du 09 Janvier. On attend ainsi plusieurs tests, une signature de fonction complète et une documentation contenant des exemples. On pourra s'inspirer du fichier de correction de cette même leçon pour avoir une idée de ce qui est attendu.
- **Librairies** On n'utilisera pas de librairie trivialisant la question (pandas par exemple) l'objectif est d'évaluer votre compétence avec le langage brut. Ceci sous peine d'être sanctionné de nombreux points.

Chaque question vaut 3 points pour un total de 15 points. La deuxième partie sera sur 25 points.

Instructions. Vous devez écrire du code python pour répondre aux questions suivantes.

1. Codez une fonction `compte_nombres` prenant en argument `a: int` et `b: int` et renvoyant le nombre d'entiers entre `a` et `b` qui sont divisibles par exactement deux nombres parmi 3, 5 et 7.
2. Codez une fonction `recupere_frequents` prenant en entrée `nombres: List[int]` et `nbr: int` et renvoyant les `nbr` nombres apparaissant le plus dans la liste. Si jamais il y a des égalités on choisira par ordre croissant. A titre d'exemple on devrait avoir

```
>>> recupere_frequents([1, 2, 1], 1)
[1]
>>> recupere_frequents([1, 1, 1, 2, 2, 4, 3], 3)
[1, 2, 3]
```
3. Codez une fonction `compte_mots`, comptant, parmi les mots ayant au plus 5 lettres (mot au sens combinaison de minuscules allant de `a` à `z` pas au sens de la langue française) le nombre qui ont deux consonnes consécutives.
4. Codez une fonction `calcule_impot` prenant en entrée `revenu: int` et retournant l'impôt sur le revenu à payer correspondant. On rappelle que les tranches sont les suivantes

```

— 0 - 10064 euros : 0%
— 10065 euros - 25659 euros : 11%
— 25660 euros - 73369 euros : 30%
— 73370 euros - 157806 euros : 41%
— 157807 euros - ... : 45%

```

A titre d'exemple on devrait avoir

```

>>> calcule_impot(10_000)
0
>>> calcule_impot(20_000)
1092
>>> calcule_impot(30_000)
3017
>>> calcule_impot(80_000)
18745

```

5. Écrire une fonction `dessine_sapin` d'argument `n: int` renvoyant une `str` dessinant un sapin de la façon suivante. Pour `n` valant 1 on récupère

```

"""
    x
   xxx
  xxxxx
 III
"""

```

pour `n` valant 2

```

"""
    x
   xxx
  xxxxx
   xxx
  xxxxx
 xxxxxxx
   III
   III
"""

```

Et ainsi de suite.