

TD 18 : Bulletins de vote

1. Génération des votes

On souhaite réaliser le comptage automatique d'un dépouillement d'une élection. Votre startup reçoit un fichier texte qui contient ligne par ligne les noms des candidats qui ont reçu le vote des électeurs. Il n'y a qu'un seul nom par ligne. Vous êtes chargé de fournir au client le nombre de votes et le total des votes reçus par candidats présents dans le fichier.

Vous allez commencer par générer vous-même une liste de votes factices à partir d'une liste de noms.

1. Allez sur python, et créez TD18-Bulletins.
2. Dans le main.py, écrire une fonction qui crée un fichier texte avec un nom par ligne qui a été tiré aléatoirement depuis une liste de noms.

```
generate_ballots(total : int, noise : int) -> bool
```

- total -- nombre de bulletins
- noise -- n° de bulletin "bruité"
- la fonction renvoie True si Ok, ou False si erreur

NB : le bulletin bruité est un vote étrangé (ie : d'un candidat pirate) qui ne fait pas partie de la liste « officielle ».

Indications : dans votre fonction :

- vous allez ouvrir en écriture un fichier "votes.txt" (que vous trouverez dans votre espace classe) ;
 - vous allez créer une liste "ballots" dans laquelle vous aurez des noms fictifs de candidats et une variable "stranger" qui contiendra le nom du candidat pirate. Si vous voulez favoriser un candidat, vous pouvez répéter plusieurs fois son nom dans la liste "ballots". Vous ajouterez un nom choisi au hasard dans la liste, sauf arrivé au moment d'écrire le candidat à la « noise_ième » position où vous écrirez dans le fichier texte le nom du candidat pirate ;
 - pour écrire un nom par ligne, vous devez concaténer (on utilise le "+" en Python pour concaténer deux chaînes de caractères) à la suite du nom du candidat la chaîne de caractères "\n".
 - vous n'oublierez pas de fermer votre fichier texte derrière vous.
3. Documenter la fonction.
 4. Vérifiez que votre fichier texte a bien été créé et qu'il y a bien un candidat par ligne.

2. Dépouillement

1. Ajouter une fonction qui permet de dénombrer le nombre de votes pour un candidat donné.

On donne le prototype de la fonction :

```
read_ballots(filename : str) -> dict
```

- filename -- nom du fichier texte
- la fonction retourne un dictionnaire sous la forme {nom du candidat:nbr de votes}, ou None si erreur

NB : On pourra utiliser les méthodes "readline()" et "rstrip()" .

2. Documenter la fonction (docstring).
3. Il y a des noms qui ont trop peu de votes pour qu'ils correspondent à des candidats. On dit que ces bulletins sont des bulletins "bruités". Déterminer ces intrus qui se sont glissés dans le fichier. On considérera qu'un bulletin est bruité s'il correspond à un candidat qui a moins de 10 votes.
4. Éliminer les intrus du dictionnaire. Pour cela, on parcourt le dictionnaire et on enregistre les clefs indésirables à supprimer par la suite.

NB : utiliser une compréhension de liste.

5. Afficher le nombre de votants à l'aide d'une compréhension de liste faite sur le dictionnaire retourné par la fonction read_ballots().

NB : on pourra utiliser la fonction "sum()".

6. Afficher le nom de chaque candidat avec le nombre de vote qu'il a eu à partir du dictionnaire retourné par la fonction read_ballots().