

THÈME 1 : Parcours séquentiel d'un tableau

Ces algorithmes simples nécessitent au maximum une boucle for. Ils correspondent donc à un parcours séquentiel de tableaux. Dans ce document, on utilise un tableau de personnes, comme proposé dans le programme appelé `theme1.py` disponible sur bouillotvincent.github.io. Une personne sera modélisée par un tuple de la forme : **(nom, age)**. Une classe est modélisée par un ensemble d'élèves, sous forme d'un tableau de tuples.

1) Algorithmes de cumul (20')

Ces algorithmes visent à agréger des données : calcul de moyenne, détermination du nombre de voyelles dans un mot etc.

Complétez la fonction `comptage` qui compte le nombre de personnes ayant moins de 16 ans strictement. Le tableau est pris comme argument d'entrée. La fonction `comptage` renvoie le nombre de personnes de moins de 16 ans.



Prolongement : Écrivez une fonction `superComptage` qui permet de faire la même chose avec une seule et unique ligne de code !

2) Algorithmes de recherche d'un extremum (30')

Ces algorithmes visent à rechercher la donnée la plus grande ou la plus petite parmi un ensemble de données.

- Complétez la fonction `preAdo` qui renvoie le nom de la personne la plus jeune du tableau. Le tableau est pris comme argument d'entrée. La fonction `preAdo` renvoie le nom du pré-ado.
- Créez une fonction `plusProcheDe` qui renvoie le nom de la personne la plus proche de l'âge de 27 ans. Si deux personnes sont les plus proches de l'âge de 27 ans, la fonction renvoie la première personne rencontrée.

3) Algorithmes de vérification (20')

Ces algorithmes ont pour but de rechercher une donnée avec une caractéristique particulière parmi un ensemble de données.

Complétez la fonction `grosFreak` qui renvoie l'indice de la première personne qui a au moins 35 ans. La fonction `grosFreak` renvoie `None` si aucune personne n'a plus de 35 ans.



Prolongement : Écrivez une fonction `superGrosFreak` qui permet de faire la même chose avec une seule et unique ligne de code !

4) Mini-projet (50')

Si vous avez le temps... Décodez le message suivant à l'aide d'un algorithme de comptage. On pourra utiliser `matplotlib.pyplot` afin de réaliser un histogramme. Un code d'exemple est proposé.

EAGHQZF, BAGD E'MYGEQD, XQE TAYYQE P'QCGUBMSQ BDQZZQZF PQE
MXNMFDAE, HMEFQE AUEQMGJ PQE YQDE,CGU EGUHQZF, UZPAXQZFE
OAYBMSZAZE PQ HAKMSQ,XQ ZMHUQD SXUEEMZF EGD XQE SAGRDRQE MYQDE.

M BQUZQ XQE AZF-UXE PQBAEQE EGD XQE BXMZOTQE,CGQ OQE DAUE PQ
X'MLGD, YMXMPDAUFE QF TAZFQGJ,XMUEEQZF BUFQGEQYQZF XQGDE SDMZPQE
MUXQE NXMZOTQEOAYYQ PQE MHUQDAZE FDMUZQD M OAFQ P'QGJ.

OQ HAKMSQGD MUXQ, OAYYQ UX QEF SMGOTQ QF HQGXQ!XGU, ZMSGQDQ EU
NQMG, CG'UX QEF OAYUCGQ QF XMUP!X'GZ MSMOQ EAZ NQO MHQO GZ NDGXQ-
SGQGXXQ,X'MGFDQ YUYQ, QZ NAUFMZ, X'UZRUDYQ CGU HAXMUF!

XQ BAQFQ QEF EQYNXMNXQ MG BDUZOQ PQE ZGQQECGU TMZFQ XM FQYBQFQ
QF EQ DUF PQ X'MDOTQD;QJUXQ EGD XQ EAX MG YUXUQG PQE TGQQE,EQE
MUXQE PQ SQMZ F X'QYBQOTQZF PQ YMDOTQD.

