

SAE 2.02

Etienne Sellier

Classement des algorithmes :

Simplicité-Meilleur :

1:

simplicité-meilleur9

note : 10/20

ne passe pas le troisième test mais très facile à lire car beaucoup de commentaires, relativement mieux que les deux autres qui ne passent aucun test

2:

simplicité-meilleur24

note : 3/20

Ne passe aucun test JUnit, 0 commentaires, je suspecte chatGPT d'être passé par là.

3:

simplicité-meilleur48

note : 0/20

Ne compile pas, 0 commentaire

Simplicité-Meilleur :

1:

simplicité-pire58

note : 13/20

Le code est respecte parfaitement la catégorie (pire simplicité) mais ne passe malheureusement pas le troisième test fourni

2:

simplicité-pire23

note : 5/20

Pas une classe Exercice, return un String et ne passe aucun des tests fournis

Efficacité-Meilleur :

1:

efficacitemeilleu-r34

note : 16/20

Ne passe malheureusement pas le test de la chaîne vide mais très rapide (3ms)

2:

efficacite-meilleur7

note : 3/20

Le code renvoie toutes les lettres de la phrase séparément, ne passe aucun test fournie

Efficacité-Pire :

1:

efficacité-pire29

note : 16/20

Passe quasiment tout les test fournie et excelle dans sa catégorie (12ms de temps d'exécution)

2:

efficacité-pire10

note : 9/20

Ne passe malheureusement pas les test Junit mais programme a peut près fonctionnel et respecte la catégorie avec un temps d'exécution de 8ms

Sobriété-Meilleur :

1:

sobriété-meilleur63

note : 12/20

Ce code ne passe que le second test fourni, son temp d'exécution est lui de 5ms et le code fonctionne en général, la façon de trier les mots est bel et biens sobre car une seul liste est utilisé et les mots sont déplacé au début de la liste au fur et à mesure des boucles for

2:

sobriété-meilleur3

note : 4/20

Ne passe pas les test fournie, renvoie :[the, the, number, of, beast], a un temp d'exécution de 13ms

Sobriété-Pire :

1:

sobriété-pire15

note : 19/20

ce code passe tout les test fourni 🥳🥳 et a un
temp d'exécution de 19ms, il utilise plusieurs
ArrayList, StringBuilder List, ce qui prend
beaucoup de place en mémoire