fonctionnalité :	Recherche de recette	**Fonctionnalité #1**

Problématique : Améliorer l'expérience utilisateur, nous cherchons à implémenter une fonctionnalité de recherche de recettes qui soit à la fois performante et facile à maintenir

Option 1 : Algorithme avec boucle for
Dans cette option, la recherche est réalisée en utilisant des boucles for...of, ce qui permet un contrôle plus direct du flux d'exécution.

Avantages :	**Inconvénients :**
Une légère amélioration de la performance peut être observée pour des collections très grandes, grâce à l'optimisation de la sortie anticipée de la boucle (break)	Plus verbeux, ce qui peut rendre le code plus long et potentiellement plus sujet à des erreurs lors de modifications futures.
Approche plus intuitive pour les développeurs	De multiples niveaux de boucles peuvent rendre le code plus complexe et difficile à maintenir.
Permet de sortir de la boucle dès qu'une condition est remplie (break), évitant ainsi des vérifications inutiles.	
Complexité temporelle : O(n * m), où n est le nombre de recettes et m est le nombre du String qui est entée dans le champs de recherche principal par recette. Chaque entrée de chaque recette est vérifié individuellement.	

Option 2 : Algorithme boucle fonctionnelle
Dans cette option, la recherche est effectuée en utilisant une approche fonctionnelle avec les méthodes filter et some de JavaScript

Avantages :	**Inconvénients :**
Le code est plus concis et utilise des méthodes d'ordre supérieur, ce qui le rend plus lisible pour les développeurs familiers avec la programmation fonctionnelle.	Peut avoir une légère surcharge en termes de performance due à l'utilisation de méthodes d'ordre supérieur
Moins de lignes de code signifient moins de risques d'erreurs et une maintenance plus facile.	Pour ceux qui ne sont pas habitués à la programmation fonctionnelle, cette approche peut sembler moins intuitive
La logique de filtrage et de recherche est clairement exprimée, ce qui rend le code plus compréhensible.	
Complexité temporelle : O(n * m), où n est le nombre de recettes et m est le nombre du String qui est entée dans le champs de recherche principal par recette. Chaque recette est vérifiée indépendamment pour le terme de recherche.	

Solution retenue : L'approche à retenir dépendra de la taille des collections de données typiques et de l'expertise des développeurs. Pour des collections de taille modérée et une équipe de développement à l'aise avec JavaScript fonctionnel, l'AlgoRechercheFonctionnel est recommandé pour sa lisibilité et sa maintenance facilitée.
Pour des collections très grandes et une équipe moins expérimentée avec la programmation fonctionnelle, l'AlgoRechercheForOf pourrait offrir une légère amélioration de la performance. C'est pourquoi **j'ai choisi l'AlgoRechercheForOf** pour le site "Les petits plats"