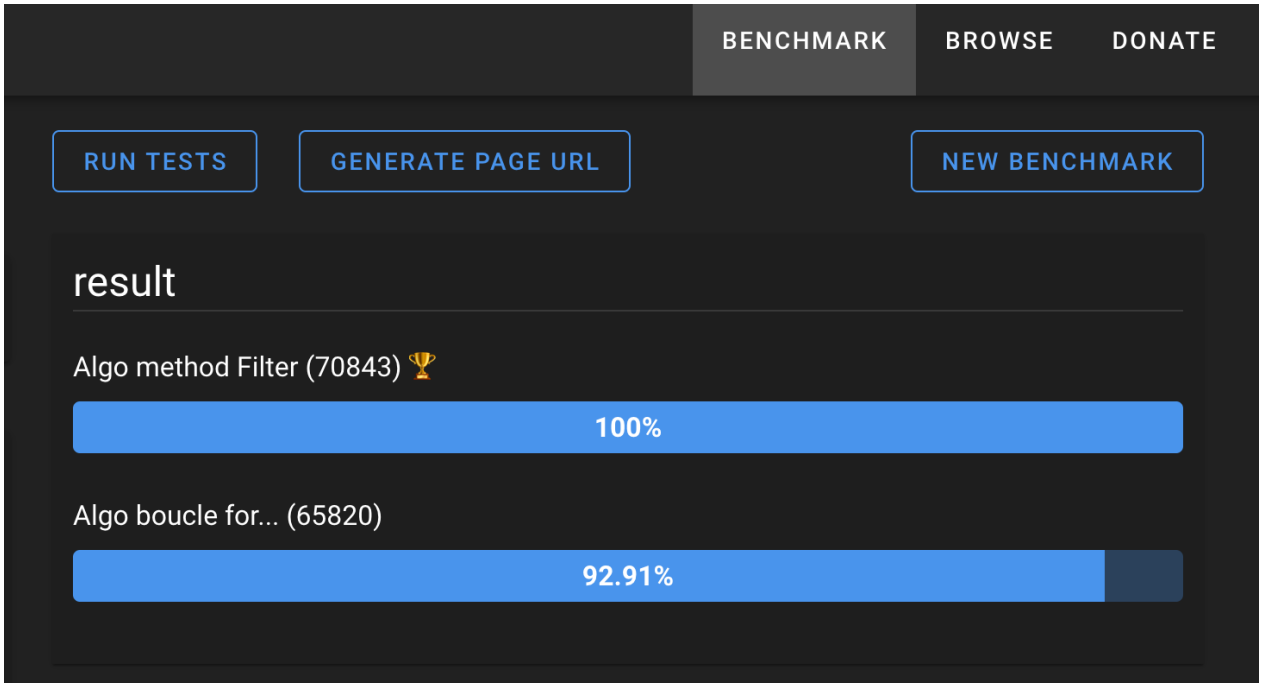
Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes	**Fonctionnalité #1**
Problématique : Filtrage des recettes dans l'interface utilisateur, l'utilisateur doit pouvoir accéder rapidement à la recette correspondant à sa recherche	

Option 1 : Programmation fonctionnelle Utilisation des méthodes de l'objet Array (forEach, Filter…) Emploi ici de la méthode « filter » qui filtre les recettes suivant la saisie effectuée et les correspondances trouvées dans le nom ou la description ou les ingrédients de la recette. La recette trouvée est ajoutée à un tableau qui servira à l'affichage des recettes. De ce tableau, les différentes listes ingrédients, ustensiles et appareils sont mises à jour.	
Avantages - code plus robuste et plus stable - moins de lignes de code - version plus performante	**Inconvénients** - code moins lisible
Saisie de 3 caractères minimum dans le champ de recherche principal	

Option 2 : Programmation native Utilisation des boucles (for …). Ici utilisation de « for » qui itère sur le tableau des recettes et cherche s'il existe une correspondance entre la saisie, et le nom ou la description ou un des ingrédients de la recette. Si oui, la recette en question est ajoutée à un nouveau tableau qui servira à l'affichage des recettes trouvées. De ce tableau, les différentes listes ingrédients, ustensiles et appareils sont mises à jour également	
Avantages - code plus lisible et plus facile à comprendre	**Inconvénients** - code moins stable, moins performant - version plus lente
Saisie de 3 caractères minimum dans le champ de recherche principal	

Solution retenue :
Notre choix se porte donc sur **l'option 1**, la programmation fonctionnelle avec **Filter**

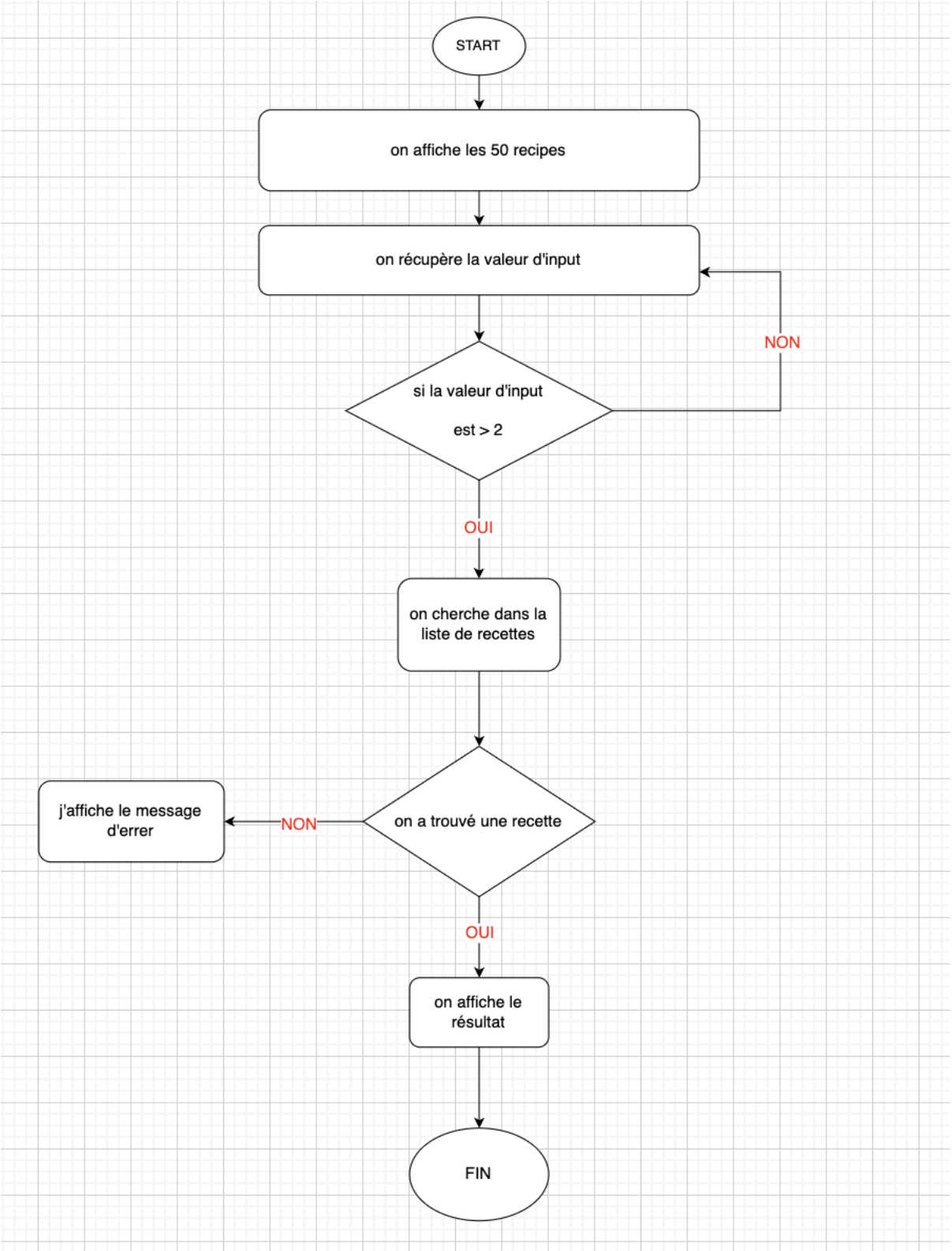


Algo method Filter

```
let result = [];
const valueInput = 'coco';
result = recipes.filter(recipe =>
        recipe.name.toLowerCase().includes(valueInput) ||
        recipe.description.toLowerCase().includes(valueInput) |
        recipe.ingredients.some(ingredient => ingredient.ingred
```

Algo boucle for... □

```
let results = [];
const valueInput = 'coco';
for(let i = 0; recipes.length > i; i++) {
        if(recipes[i].name.toLowerCase().includes(valueInput)
           recipes[i].description.toLowerCase().includes(valueI
             results.push(recipes[i]);
          } else {
            for(let y = 0; recipes[i].ingredients.length > y; y
              if(recipes[i].ingredients[y].ingredient.toLowerCa
                results.push(recipes[i]);
                break;
              }
```


START
on affiche les 50 recipes
on récupère la valeur d'input
NON
si la valeur d'input
est > 2
OUI
on cherche dans la liste de recettes
j'affiche le message d'errer
NON
on a trouvé une recette
OUI
on affiche le résultat
FIN