

Fiche d'investigation de fonctionnalité

Fonctionnalité : Filtre / Recherche principale	Fonctionnalité #1
Problématique : Accéder rapidement à une recette correspondant à un besoin de l'utilisateur dans les recettes déjà reçues	

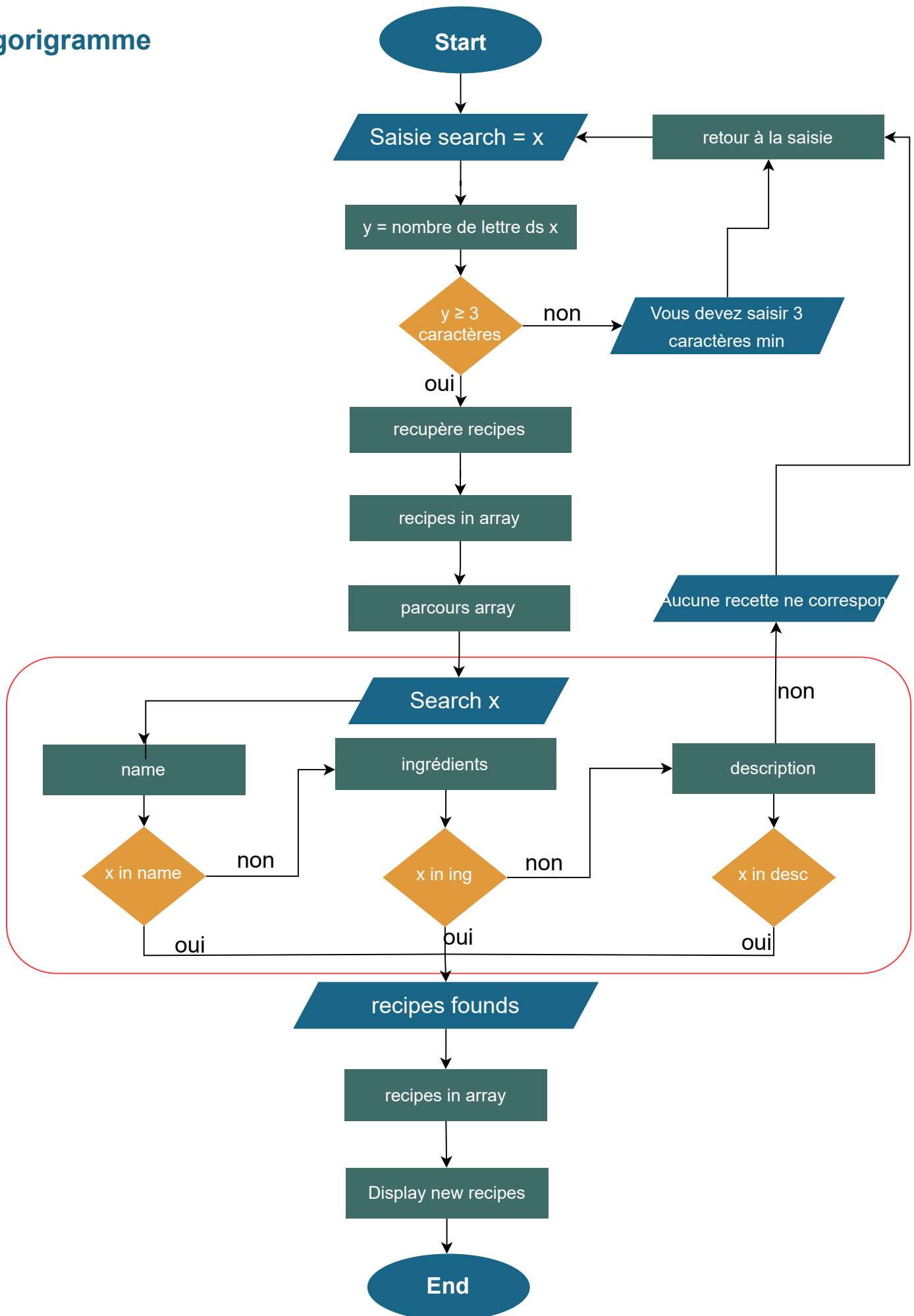
Option 1 : programmation fonctionnelle (cf Annexe1) Dans cette option, nous utilisons la méthode filter qui filtre les recettes suivant la saisie effectuée et les résultats trouvés dans le nom, le matériel, la description ou les ingrédients de la recette. Si oui, la recette en question est ajoutée à un nouveau tableau qui servira à l'affichage des recettes trouvées.	
Avantages <ul style="list-style-type: none">⊕ Code robuste et maintenable⊕ Dry⊕ version plus rapide	Inconvénients <ul style="list-style-type: none">⊖ moins de lisibilité dans les étapes d'exécution.
Saisie de 3 caractères minimum dans le champs de recherche principal	

Option 2 : Programmation native (cf annexe 2) 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 et la mise à jour des listes déroulantes.	
Avantages: <ul style="list-style-type: none">⊕ code plus lisible, plus facile à comprendre⊕ code stable	Inconvénients <ul style="list-style-type: none">⊖ version plus lente surtout en cas d'augmentation des données
Saisie de 3 caractères minimum dans le champs de recherche principal	

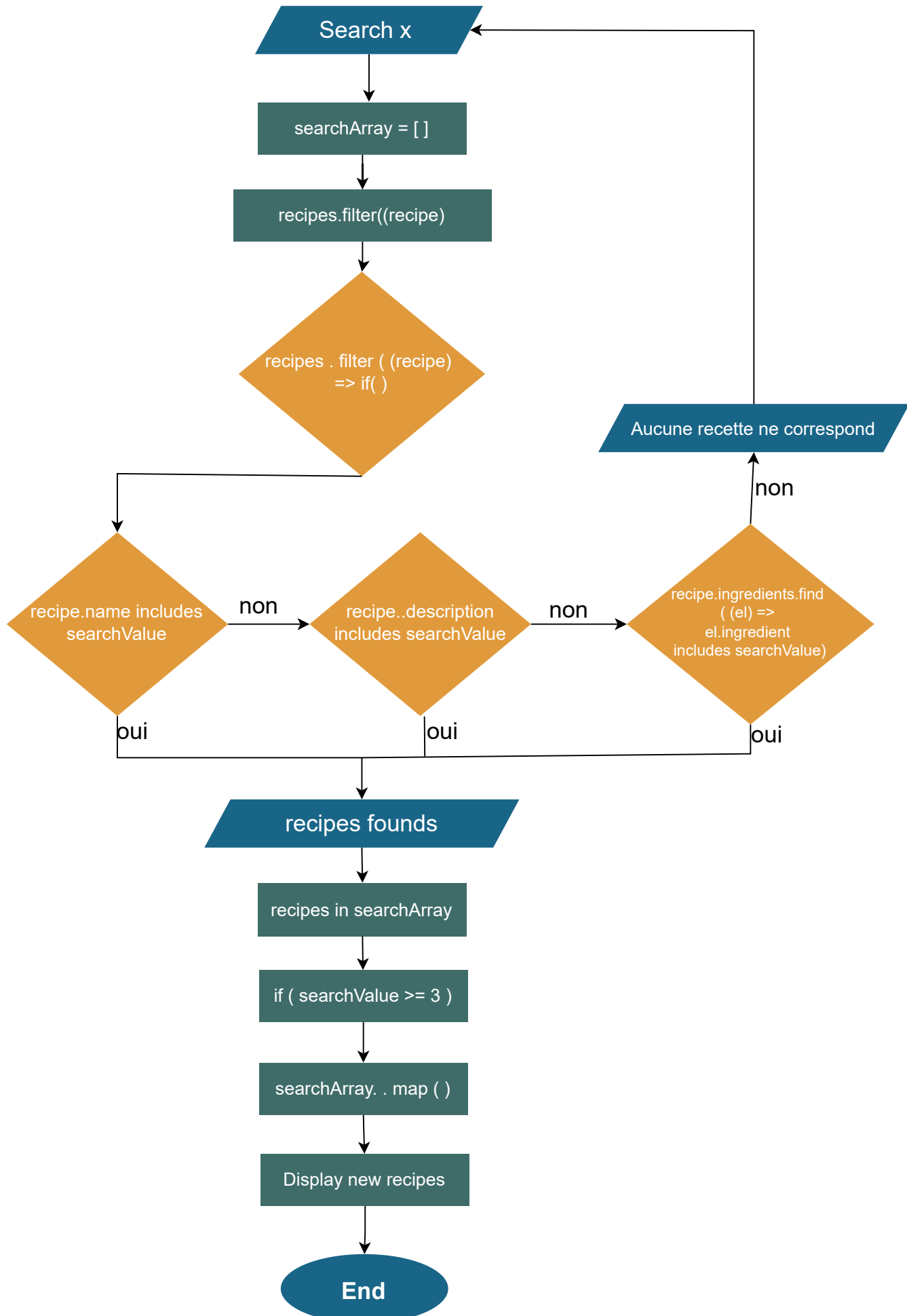
Solution retenue : Test avec jsbench.me: <ul style="list-style-type: none">- la programmation fonctionnelle avec "filter" et "include" est la plus rapide d'après le test jsbench.me (Annexe 3)- la programmation fonctionnelle est 1.25% plus rapide- Par contre la programmation native s'avère plus performante si on souhaite prendre en compte la maintenabilité du code
--



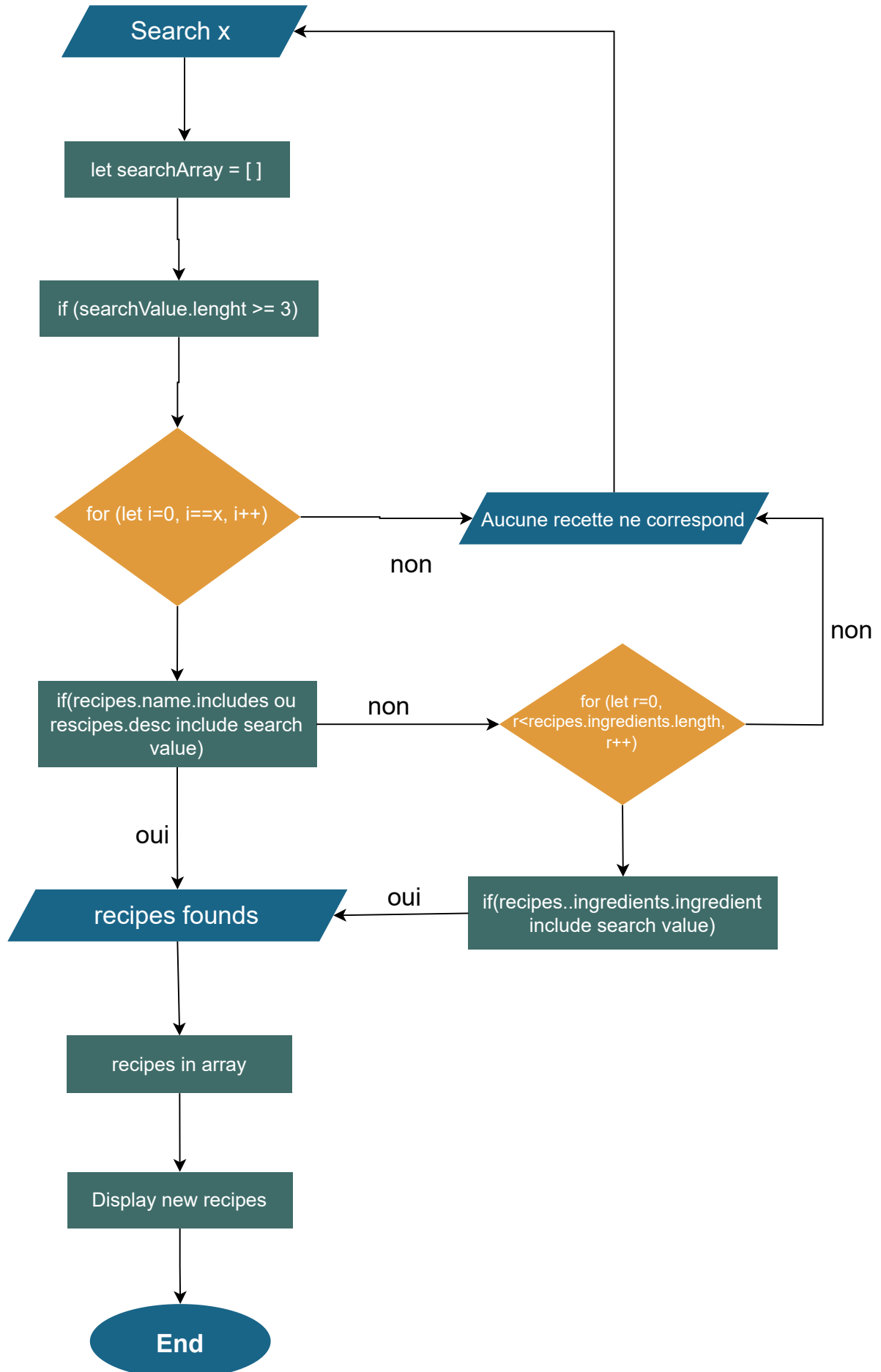
Algorithme



Annexe 1 : Diagramme de programmation fonctionnelle

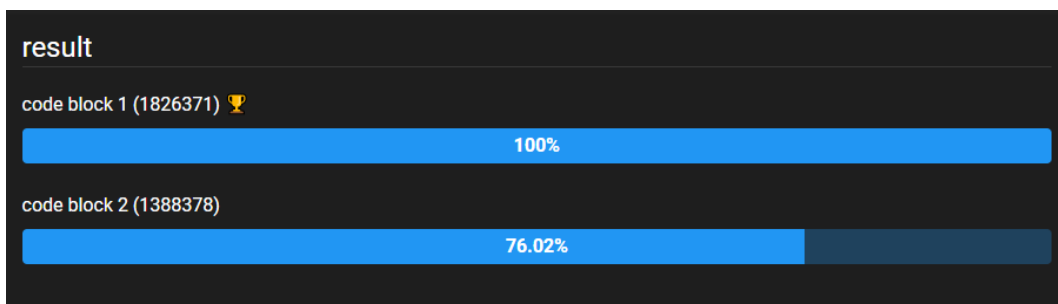
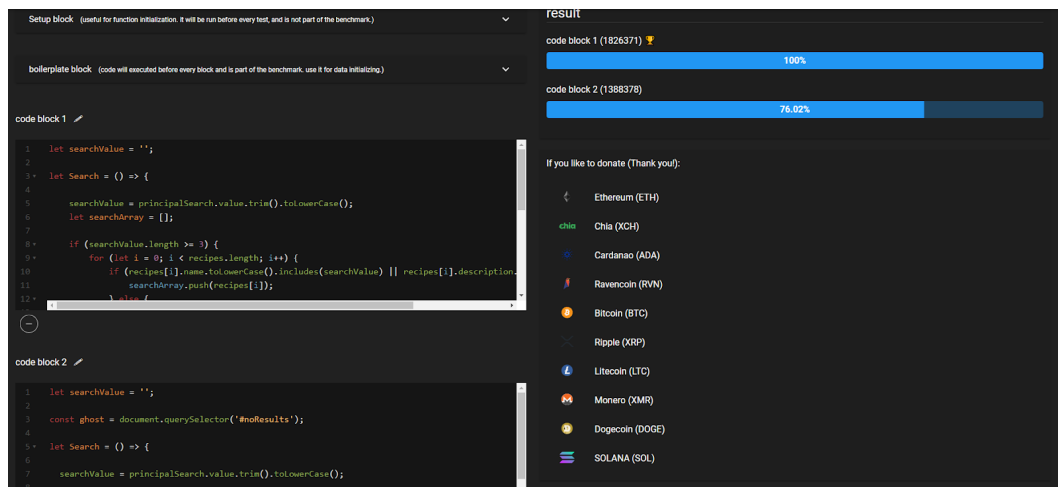


Annexe 2 : Diagramme de programmation native



Annexe 3 - Résultats des test de performance

Résultats avec jsben.ch



code block 1

```

5   searchValue = principalSearch.value.trim().toLowerCase();
6   let searchArray = [];
7
8   if (searchValue.length >= 3) {
9     for (let i = 0; i < recipes.length; i++) {
10      if (recipes[i].name.toLowerCase().includes(searchValue) || recipes[i].description.
11        searchArray.push(recipes[i]);
12    } else {
13      for (let r = 0; r < recipes[i].ingredients.length; r++) {
14        if (recipes[i].ingredients[r].ingredient.toLowerCase().includes(searchValue)
15          searchArray.push(recipes[i]);
16        break;

```

code block 2

```

1 let searchValue = '';
2 const ghost = document.querySelector('#noResults');
3 let Search = () => {
4   searchValue = principalSearch.value.trim().toLowerCase();
5   const searchArray = [];
6   recipes.filter((recipe) => {
7     if (recipe.name.trim().toLowerCase().includes(searchValue) || recipe.description.trim().to
8     el.ingredient.trim().toLowerCase().includes(searchValue))) {
9     searchArray.push(recipe);
10   }
11 });
12 }

```

[Run](#)
[Please login/register to save & publish tests](#)

[Sponsor](#)
[Settings](#)
[Sign in](#)

v1
by

Setup HTML - click to add setup HTML

Setup JS - click to add setup JavaScript

<div>enter test case name</div> <div>finished</div> <div>631375827.98 ops/s ± 0.93%</div> <div>1.25 % slower</div>	<pre> let search = () => { searchValue = principalSearch.value.trim().toLowerCase() let searchArray = [] if (searchValue.length >= 3) { for (let i = 0; i < recipes.length; i++) { if (recipes[i].name.toLowerCase().includes(searchValue) recipes[i].description.includes(searchValue)) { searchArray.push(recipes[i]) } else { for (let r = 0; r < recipes[i].ingredients.length; r++) { if (recipes[i].ingredients[r].toLowerCase().includes(searchValue)) { searchArray.push(recipes[i]) } } } } } return searchArray } </pre>	<input type="checkbox"/> DEFER
<div>enter test case name</div> <div>finished</div> <div>639353067.99 ops/s ± 0.38%</div> <div>Fastest</div>	<pre> let search = () => { searchValue = principalSearch.value.trim().toLowerCase() const searchArray = [] recipes.filter((recipe) => { if (recipe.name.toLowerCase().includes(searchValue) recipe.description.toLowerCase().includes(searchValue) recipe.ingredients.find((el) => { el.ingredient.trim().toLowerCase().includes(searchValue) }) { searchArray.push(recipe) } }) return searchArray } </pre>	<input type="checkbox"/> DEFER