Solutions détaillées des problèmes

# Problème 1 : Somme des éléments distincts

## Méthode détaillée

Étape 1 : Initialisation  
- Initialiser une variable pour la somme à 0.  
somme = 0  
  
Étape 2 : Vérification des éléments du premier ensemble  
- Parcourir chaque élément du premier tableau.  
- Pour chaque élément, vérifier s’il n’existe pas dans le deuxième tableau. Si oui, l'ajouter à la somme.  
  
Étape 3 : Vérification des éléments du deuxième ensemble  
- Parcourir chaque élément du deuxième tableau.  
- Pour chaque élément, vérifier s’il n’existe pas dans le premier tableau. Si oui, l'ajouter à la somme.  
  
Exemple détaillé :  
Ensemble1 = [3, 1, 7, 9], Ensemble2 = [2, 4, 1, 9, 3]  
- 3 présent dans Ensemble2 ? Oui -> Non ajouté  
- 1 présent dans Ensemble2 ? Oui -> Non ajouté  
- 7 présent dans Ensemble2 ? Non -> ajouté (somme = 7)  
- 9 présent dans Ensemble2 ? Oui -> Non ajouté  
- 2 présent dans Ensemble1 ? Non -> ajouté (somme = 7 + 2 = 9)  
- 4 présent dans Ensemble1 ? Non -> ajouté (somme = 9 + 4 = 13)  
Résultat final : somme = 13

## Complexité

Complexité en temps : O(n²), chaque élément est comparé à tous les autres.

# Problème 2 : Produit scalaire et orthogonalité

## Méthode détaillée et calculs utilisés

### Calcul du produit scalaire (dot\_product)

Le produit scalaire de deux vecteurs v1 et v2 de même dimension n est défini par :  
v1 · v2 = Σ(v1[i] × v2[i]), avec i allant de 1 à n.  
  
Exemple : v1 = [1, 3, -5], v2 = [4, -2, -1]  
Produit scalaire : (1×4) + (3×-2) + (-5×-1) = 4 - 6 + 5 = 3

### Vérification de l’orthogonalité

Deux vecteurs sont orthogonaux si leur produit scalaire est nul.  
Exemple : v1 = [1, 0, -1], v2 = [1, 5, 1]  
Produit scalaire : (1×1) + (0×5) + (-1×1) = 1 + 0 - 1 = 0 → Orthogonaux.

### Modification en fonction

Fonction dot\_product(v1, v2):  
 ps = 0  
 Pour i allant de 1 à taille(v1):  
 ps = ps + v1[i] × v2[i]  
 retourner ps