

Niveau 1 : Bases

1. **Afficher les éléments** Écris un programme qui affiche un à un les éléments d'une liste donnée.

Exemple : `[2, 10, 15]` →

```
2
10
15
```

2. **Somme des éléments** Demande à l'utilisateur de saisir plusieurs nombres (dans une liste) et affiche leur somme.

Niveau 2 : Manipulation simple

3. **Trouver le maximum et le minimum** Demande une liste de nombres et affiche la plus grande et la plus petite valeur.
4. **Inverser une liste** Écris un programme qui affiche une liste inversée. Exemple : `[10, 20, 30]` → `[30, 20, 10]`
5. **Compter les occurrences** Compte combien de fois une valeur donnée apparaît dans une liste. Exemple : `[2, 4, 2, 5, 2]` et valeur `2` → 3 fois.

Niveau 3 : Exercices un peu plus avancés

6. **Positions d'un élément** Affiche toutes les positions (indices) où se trouve une valeur donnée dans une liste. Exemple : `[1, 2, 3, 2, 4, 2]` et valeur `2` → positions 1, 3, 5.
7. **Moyenne des valeurs** Calcule et affiche la moyenne des éléments d'une liste de nombres.
8. **Séparer pairs et impairs** À partir d'une liste de nombres, crée deux nouvelles listes :
- une avec les nombres pairs
 - une avec les nombres impairs

Niveau 4 : Plus difficiles

9. **Fusion et tri** Demande deux listes à l'utilisateur, fusionne-les et affiche la liste triée.
10. **Doublons** À partir d'une liste, crée une nouvelle liste sans doublons. Exemple : `[1, 2, 2, 3, 1, 4]` → `[1, 2, 3, 4]`.
11. **Produit des éléments** Écris un programme qui calcule le produit de tous les éléments d'une liste.
12. **Rotation de liste** Décale tous les éléments d'une liste d'un rang vers la droite. Exemple : `[1, 2, 3, 4]` → `[4, 1, 2, 3]`.