

# Différence entre sommes directes et inversées

## Description :

La somme  $S_n = \sum_{k=1}^n \frac{1}{k}$  peut être calculée :

- Dans l'ordre croissant  $(1, 1/2, 1/3, \dots)$ .
- Dans l'ordre décroissant  $(1/n, 1/(n-1), \dots, 1)$ .

## Objectif :

- Comparer les résultats obtenus dans les deux ordres pour observer l'effet des erreurs d'arrondi.

## Étapes :

- Initialisez deux variables pour les sommes ( $S_{\text{croissant}}$  et  $S_{\text{décroissant}}$ ).
- Utilisez une boucle pour calculer  $S_{\text{croissant}}$ .
- Utilisez une boucle pour calculer  $S_{\text{décroissant}}$ .
- Affichez les deux résultats.

## Code :

```
#include <stdio.h>

int main() {
    int n = 10; // Nombre de termes
    double S_croissant = 0.0, S_decroissant = 0.0;

    // Étape 2 : Somme croissante
    for (int i = 1; i <= n; i++) {
        S_croissant += 1.0 / i;
    }

    // Étape 3 : Somme décroissante
    for (int i = n; i >= 1; i--) {
        S_decroissant += 1.0 / i;
    }

    // Étape 4 : Affichage des résultats
    printf("Somme croissante : %lf\n", S_croissant);
    printf("Somme décroissante : %lf\n", S_decroissant);

    return 0;
}
```

# Ratio d'une suite proche de ( $\phi$ )

## Description :

La suite de Fibonacci est définie par :

- $b_0 = 1, b_1 = 1$
- $b_{n+2} = b_{n+1} + b_n$

Les ratios successifs  $r_n = \frac{b_{n+1}}{b_n}$  convergent vers le nombre d'or  $\phi = \frac{1+\sqrt{5}}{2}$ .

## Objectif :

- Calculer les 20 premiers termes de la suite.
- Calculer et afficher les ratios successifs.
- Comparer les ratios avec  $\phi$ .

## Étapes :

- Déclarez un tableau pour stocker les termes de Fibonacci.
- Initialisez  $b_0 = 1$  et  $b_1 = 1$ .
- Utilisez une boucle pour calculer les 20 premiers termes.
- Calculez les ratios successifs.
- Affichez les termes et les ratios.

## Code :

```
#include <stdio.h>
#include <math.h>

int main() {
    int n = 20; // Nombre de termes
    int b[21]; // Tableau pour les termes de Fibonacci
    double ratios[20]; // Tableau pour les ratios

    // Étape 2 : Termes initiaux
    b[0] = 1;
    b[1] = 1;

    // Étape 3 : Calcul des termes de Fibonacci
    for (int i = 2; i <= n; i++) {
        b[i] = b[i-1] + b[i-2];
    }

    // Étape 4 : Calcul des ratios successifs
    for (int i = 1; i <= n; i++) {
        ratios[i-1] = (double)b[i] / b[i-1];
    }

    // Étape 5 : Affichage
    printf("Les termes de Fibonacci sont :\n");
    for (int i = 0; i <= n; i++) {
        printf("b(%d) = %d\n", i, b[i]);
    }

    printf("\nLes ratios successifs sont :\n");
    for (int i = 0; i < n; i++) {
        printf("b(%d)/b(%d) = %lf\n", i+1, i, ratios[i]);
    }

    // Comparaison avec le nombre d'or
    printf("\nValeur théorique du nombre d'or : %lf\n", (1 + sqrt(5)) / 2);

    return 0;
}
```

## Points communs dans tous les exercices :

- Stockage** : Utilisation de tableaux statiques pour enregistrer les termes ou les résultats intermédiaires.
- Boucles** : Utilisation de boucles `for` pour calculer les termes récurrents ou accumuler des sommes.
- Affichage** : Vérification des résultats étape par étape grâce à des `printf`.