Exo :

Exercice 1 :

- Écrivez une fonction multiply qui prend deux nombres en argument et retourne leur multiplication. Ensuite, utilisez cette fonction pour calculer la multiplication de 4 et 6.
- Écrivez une fonction divide qui prend deux nombres en argument et retourne leur division. Ensuite, utilisez cette fonction pour calculer la division de 10 par 2.
- Créez une fonction operate qui prend deux nombres et une fonction opération (comme add, multiply, divide, etc.) en argument et retourne le résultat de l'opération sur ces nombres.

Exo :

Exercice 2 :

- Écrivez une fonction pureAdd qui prend deux nombres en argument et retourne leur somme sans utiliser de variable globale.
- Écrivez une fonction pureSubtract qui prend deux nombres en argument et retourne leur différence sans utiliser de variable globale.
- B. Modifiez la fonction pureSubtract pour qu'elle puisse accepter un nombre variable d'arguments et renvoie la soustraction de tous ces nombres.

Exo :

Exercice 3 :

- Écrivez une fonction higherOrderFunction qui prend une fonction en argument et retourne une nouvelle fonction qui double le résultat de la fonction donnée.
- Écrivez une fonction mapToUpper qui prend un tableau de chaînes de caractères en argument et retourne un nouveau tableau avec toutes les chaînes en majuscules.
- Utilisez mapToUpper pour transformer un tableau de mots en un tableau où chaque mot est en majuscule.

Exo :

Exercice 4 :

- À partir d'un tableau de nombres, utilisez map pour obtenir un nouveau tableau où chaque élément est le carré de l'élément d'origine. Ensuite, utilisez filter pour obtenir un tableau contenant uniquement les nombres pairs de ce tableau. Enfin, utilisez reduce pour trouver la somme de ces nombres pairs.
- À partir d'un tableau de nombres, utilisez filter pour obtenir un tableau contenant uniquement les nombres impairs de ce tableau.
- Utilisez reduce pour trouver le produit des nombres impairs dans le tableau.

Exercice 5 :

- Écrivez une fonction récursive factorielle(n) qui prend un entier positif n en paramètre et retourne sa factorielle.

La factorielle d'un nombre n (notée n!) est le produit de tous les entiers positifs de 1 à n.

Exemple :

factorielle(5) devrait retourner 120 car 5! = 5 * 4 * 3 * 2 * 1 = 120.

Instructions :

Utilisez une fonction récursive pour calculer la factorielle de n.

Gérez le cas de base où n est égal à 0 ou 1.

Assurez-vous que la fonction factorielle gère correctement les cas où n est un nombre entier positif.

Exercice 6 :

- Écrivez une fonction récursive sommeChiffres(n) qui prend un nombre entier positif n en paramètre et retourne la somme de ses chiffres.

Exemple :

sommeChiffres(123) devrait retourner 6 car 1 + 2 + 3 = 6.

Instructions :

- Utilisez une fonction récursive pour calculer la somme des chiffres de n.
- Gérez le cas de base où n est un nombre à un seul chiffre.

Exercice 6 :

- Écrivez une fonction récursive fibonacci(n) qui prend un entier positif n en paramètre et retourne le n-ième terme de la séquence de Fibonacci. La séquence de Fibonacci commence par 0 et 1, et chaque terme suivant est la somme des deux termes précédents.

Exemple :

fibonacci(6) devrait retourner 8 car le 6ème terme de la séquence de Fibonacci est 8 (0, 1, 1, 2, 3, 5, 8).

Instructions :

- Utilisez une fonction récursive pour calculer le n-ième terme de la séquence de Fibonacci.
- Gérez le cas de base pour les deux premiers termes de la séquence (n égal à 0 ou 1).