

Exercices

1. Réalisez une fonction prenant le nom et l'âge de la personne en entrée et retournant une chaîne de caractères "{nom} a {age} ans";
2. Réalisez une fonction qui retourne vrai si le nombre entré est pair et faux s'il est impair
3. Réalisez une fonction qui prend un tableau d'entiers et qui renvoie le tableau inversé
4. Réalisez une fonction qui calcule combien de fois un nombre donné est présent dans un tableau donné et renvoie cette valeur
5. Pareil que la 4 mais pour une suite de nombre
6. Réalisez les fonctions nécessaires pour faire les 4 opérations (+ - / *) avec 2 nombres donnés
7. Réalisez une fonction permettant de choisir laquelle des 4 fonctions réalisées au point 6 appeler dépendant de l'entrée d'un utilisateur
8. Réalisez les fonctions nécessaire pour jouer au jeu de Nim (bâtonnets)
9. Réalisez les fonctions nécessaire pour renvoyer la somme des valeurs d'un tableau

Exercices avancés

1. Réalisez une fonction récursive permettant de calculer la factorielle d'un nombre (factorielle de 5 = $5*4*3*2*1$, factorielle de 7 = $7*6*5*4*3*2*1$ etc)
2. Réalisez un ensemble de fonctions permettant de vérifier si une valeur entrée dans une grille de 4 par 4 respecte les règles du sudoku (voir grille plus bas)
3. Réalisez les fonctions nécessaires pour permettre de remplir les trous de la grille
4. Réalisez un générateur de grilles de sudoku de 4 par 4
5. Réalisez le tout nécessaire pour générer et compléter des sudoku de 4 par

//exemple de grille de 4 par 4 à trou (les 0 = pas de réponse)

```
let grid = [  
    [2, 0, 0, 0],  
    [1, 0, 0, 3],  
    [4, 0, 0, 0],  
    [3, 2, 0, 1]  
]
```

Conseil : renseignez vous sur les backtracking algorithms pour résoudre ce genre de problème