

Fonctions et Fichiers
Travaux dirigés et pratiques 4
Algorithmique et Programmation
ASI5

Exercice 1 :

Ecrire, en utilisant deux méthodes différentes, une fonction qui calcule la somme des chiffres d'un nombre entier strictement positif et recommence le calcul avec le résultat obtenu tant que celui-ci n'est pas compris entre 1 et 9. Après chaque calcul la fonction affiche à l'écran la somme obtenue. La fonction retournera le nombre entre 1 et 9 obtenu

Exercice 2 :

La suite des nombres de Fibonacci est définie par ses deux premiers termes

$$F_0 = 0, F_1 = 1$$

$$\forall n \geq 0 \quad F_{n+2} = F_{n+1} + F_n$$

- 1: Réalisez une fonction nommée *fibonacci* paramétrée par un entier n qui calcule le nombre F_n .
- 2: Dessinez l'arbre des appels à la fonction *fibonacci* dans le calcul de *fibonacci(4)*.
- 3: Trouver une relation de récurrence pour le nombre d'appels à la fonction *fibonacci* pour calculer F_n . Puis programmez le calcul de ce nombre.

Exercice 3 :

Ecrire un programme qui permet de faire deviner un nombre choisi aléatoirement entre 0 et 100 (fonction random : $nbatrouver=(rand()%(max+1))+min;$, de la librairie stdlib). Le programme signalera à chaque tentative si le nombre proposé est plus petit ou plus grand que la solution.

Exercice 4 :

Ecrire une fonction qui reçoit comme paramètre une liste d'entiers dans un tableau d'entiers et parcourt cette liste en comptant les répétitions successives des valeurs. La fonction retournera un tableau à deux colonnes dont la première contient les valeurs de la liste dans leur ordre d'apparition et la seconde leur nombre de répétitions successives.

Exercice 5 :

1. Ecrire des fonctions qui reçoivent comme paramètre une liste d'entiers dans un tableau d'entiers et effectuent le tri du tableau selon les algorithmes de tri vues en cours
2. Ecrire un programme qui demande à l'utilisateur quel algorithme utiliser avant de trier le tableau dont les valeurs sont saisies au clavier.