SÉRIE N° 17 : FONCTIONS

CRÉATION ET UTILISATION (APPEL) D'UNE FONCTION

PARTIE I : Ecrire l'en-tête (prototype) et un exemple d'appel de chaque fonction :

1. Écrire la fonction ***factorielle*** qui prend en paramètre un entier **n** et qui retourne la factorielle du nombre **n**.
2. Écrire la fonction ***EstDivisible*** qui prend en argument deux entiers a et b et qui retourne **1** si **a** est divisible par **b**, et **0** sinon.
3. Écrire la fonction ***ChangeSigne*** qui change le signe de tous les éléments d'un tableau T d'entiers de taille N passé en paramètre.
4. Écrire la fonction ***Moyenne*** qui prend en paramètre un tableau de nombre flottants de taille N et qui retourne la valeur moyenne des éléments du tableau.
5. Écrire la fonction ***Recherche*** qui prend en paramètre un tableau d'entiers de taille N et un entier V et qui cherche et retourne la position de V dans le tableau ou -1 si V n'existe pas dans le tableau.
6. Écrire la fonction ***Maximum*** qui retourne la valeur de l'élément le plus grand d'un tableau T de nombre flottants de taille N passé en paramètre.
7. Écrire la fonction ***PosMax*** qui retourne la position de la valeur de l'élément le plus grand d'un tableau T de nombre flottants de taille N passé en paramètre.
8. Écrire la fonction ***supprime_espace*** qui prend en argument une chaîne de caractères **S** et qui supprime les espaces et les tabulations d'une chaîne de caractères **S**. Cette fonction ne doit pas utiliser de tableau intermédiaire.
9. Écrire la fonction **Inverse** qui inverse une chaîne de caractères **S** et met le résultat dans une autre chaîne **D**. Les chaînes sont données en argument. Utiliser la fonction **strlen.**
10. Ecrire la fonction ***min2maj*** qui prend en argument une chaîne de caractères **S** et qui permet de changer toutes les lettres minuscules d'une chaîne de caractères **S** en majuscules.
11. Écrire une fonction ***compare*** qui prend en argument deux chaînes de caractères ***ch1*** et ***ch2*** et qui compare les deux chaînes ***ch1*** et ***ch2*** et retourne une valeur négative, nulle ou positive

selon que ***ch1*** est lexicographiquement inférieure, égale ou supérieure à ***ch2***. (**Sans utiliser strcmp**)

12. Ecrire une fonction ***palindrome*** qui prend en paramètre une chaîne de caractère et qui teste si la chaîne est un palindrome.

On appelle palindrome une suite de caractères qui se lit de la même façon dans les deux sens.

exemple: ''laval'', ''ressasser ''.

13. Ecrire la fonction ***compte_mots*** qui prend en paramètre une chaîne de caractère **phrase** et qui retourne le nombre de mots dans une phrase (on considérera que les mots ne sont séparés que par un seul espace, et ne contient pas de ponctuation).

Exemple :

phrase: "le langage c est un langage de programmation"

Cette phrase contient 8 mots

14. Ecrire la fonction **Tri_insertion** qui prend en paramètre un tableau d'entiers de taille N et qui permet de trier le tableau en ordre croissant en utilisant l'algorithme de tri par insertion

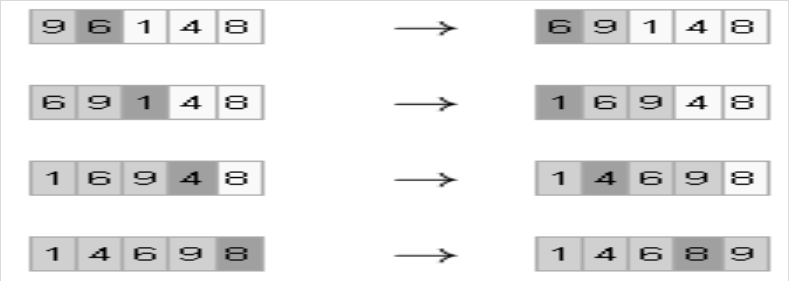
Principe :

Le tri par insertion est un algorithme de tri classique dont le principe est très simple. C'est le tri que la plupart des personnes utilisent naturellement pour trier des cartes : prendre les cartes mélangées une à une sur la table, et former une main en insérant chaque carte à sa place.

L'algorithme principal du tri par insertion est un algorithme qui insère un élément dans une liste d'éléments déjà trié

Exemple : Voici les étapes de l'exécution du tri par insertion sur le tableau $T = [9, 6, 1, 4, 8]$.

Le tableau est représenté au début et à la fin de chaque itération.



.

PARTIE II : Ecrire le corps de chaque fonction de la partie précédente

Prof : ZBAKH ABDEL ALI