

TP3 – Programmation en Langage C

Tableaux, Pointeurs et Fonctions

Exercice 1 :

- 1) Ecrire une fonction qui écrit la mention obtenue au baccalauréat correspondant à une note donnée.
- 2) Ecrire un programme dans lequel :
 - vous déclarez deux variables globales entières A et B,
 - vous définissez une fonction **Lecture** qui ne prend pas d'argument et fait la lecture de A et B,
 - vous définissez les fonctions **Somme**, **Produit**, qui prennent chacune deux arguments de type entier et retournent respectivement la somme et le produit.
 - un programme principal qui, en utilisant ces fonctions, lit A et B et affiche leur somme et leur produit.
- 3) Ecrire un programme dans lequel :
 - deux fonctions **Min** et **Max** prennent chacune deux arguments de type entier et retournent respectivement le minimum et le maximum de leurs arguments
 - Donner un exemple d'appel
- 4) Ecrire un programme dans lequel :
 - une fonction **Carre** prend un argument de type entier et le remplace par son carré.
 - Donner un exemple d'appel

Exercice 2 :

- 1) Ecrire sous forme de fonctions, un programme C qui comporte les fonctions suivantes (utilisant le formalisme Pointeur) :
 - Une fonction **Lire** qui lit un tableau de N entiers
 - Une fonction **Afficher** qui affiche un tableau de N entiers
 - Une fonction **Inverser** qui inverse l'ordre des éléments d'un tableau de N entiers

- Une fonction **Trier** qui trie un tableau de N entiers
- Une fonction **Premier** qui vérifie si un tableau de N entiers est premier. Un tableau est dit Premier si et seulement si tous ses éléments sont des nombres premiers.
- Une fonction **Rechercher** qui cherche l'existence d'un élément dans un tableau de N entiers et retourne sa première position ou -1 si l'élément est inexistant.
- Une fonction **Rechercher_Par_Dichotomie** qui utilise la méthode par dichotomie et cherche l'existence d'un élément dans un tableau de N entiers et retourne sa première position ou -1 si l'élément est inexistant. Cette fonction doit faire appel à la fonction **Trier** pour trier le tableau.
- Une fonction **Concatener** qui ajoute un tableau de M entiers à la fin d'un tableau de N entiers (utiliser la fonction *realloc*)

2) Donner un exemple d'appel

3) Modifier la fonction main de sorte à proposer à l'utilisateur un menu textuel lui permettant de choisir la fonction à appeler.

4) Modifier la fonction **Trier** de telle sorte à faire passer un paramètre indiquant l'ordre de tri désiré (0 : ordre décroissant, 1 : ordre croissant)

Exercice 3 :

1) Ecrire, sous forme de fonctions, un programme C qui implémente certaines fonctions classiques (implémentées dans string.h) de gestion des chaînes de caractères :

- **char *strcpy(char *dest, const char *src)** : copies the string pointed to by src to dest.
- **char *strncpy(char *dest, const char *src, size_t n)** : copies up to n characters from the string pointed to by src to dest. (*size_t* : This is the unsigned integral type and is the result of the *sizeof* keyword).
- **int strcmp(const char *str1, const char *str2)** : compares the string pointed to by str1 to the string pointed to by str2.
- **char *strncat(char *dest, const char *src, size_t n)** : appends the string pointed to by src to the end of the string pointed to by dest up to n characters long.

2) Donner des exemples d'appel

Exercice 4 :

- 1) Ecrire une fonction qui prend en argument une chaîne de caractères représentant un entier en décimal et retourne l'entier équivalent. ("1234" ---> 1234). On supposera que la chaîne est correcte et représente bien un entier. (Prototype : int chaîne_vers_entier (char *s))
- 2) Ecrire une fonction qui prend en argument une chaîne de caractères, la renverse sur elle-même ("toto"—→"otot") et retourne l'adresse de cette chaîne. (Prototype : char * miroir (char *s)).
- 3) Ecrire une fonction qui recherche dans une chaîne chaque caractère **C** pour le remplacer par un caractère **R** et retourne l'adresse de la chaîne. (Prototype : char * cherche_remplace (char c, char r, char * s)).
- 4) Donner des exemples d'appel

Exercice 5 :

- 1) Ecrire, sous forme de fonctions, un programme C qui comporte les fonctions suivantes :
 - Une fonction **Lire** qui lit une matrice d'entiers de N lignes et M colonnes.
 - Une fonction **Afficher** qui affiche une matrice d'entiers de N lignes et M colonnes.
 - Une fonction **Somme** qui calcule la somme de deux matrices
 - Une fonction **Produit** qui calcule le produit de deux matrices
 - Une fonction **Nulle** qui vérifie si une matrice d'entiers de N lignes et M colonnes est nulle (tous les éléments sont égaux à zéro)
 - Une fonction **Rechercher** qui cherche l'existence d'un élément dans une matrice d'entiers de N lignes et M colonnes et retourne sa première position ou -1 si l'élément est inexistant.
- 2) Ecrire une fonction main qui propose à l'utilisateur un menu textuel permettant de choisir la fonction à appeler