

# TP

## LES POINTEURS

### Exercice 1

Ecrire de deux façons différentes, un programme qui lit 10 nombres entiers dans un tableau avant d'en chercher le plus grand et le plus petit

- Q1. En utilisant uniquement le formalisme tableau
- Q2. En utilisant le formalisme pointeur à chaque fois que cela est possible

### Exercice 2

- Q1. Ecrire une fonction d'entête `float rand01(void)` renvoie un réel aléatoire dans l'intervalle  $[0, 1[$ . Utiliser la fonction `rand()` de la bibliothèque `stdlib`.
- Q2. Ecrire une fonction d'entête `int myRand(int n)` renvoie un entier aléatoire dans l'intervalle  $[0, n[$ .
- Q3. Ecrire une fonction d'entête `void printTab(int *tab, int lg)` affiche un tableau d'entiers. Est-il possible d'écrire une fonction d'affichage générique capable d'afficher n'importe quel tableau et pas seulement un tableau d'entiers ?
- Q4. Ecrire une fonction d'entête `void printMat(int **mat, int n, int m)` affiche une matrice  $n \times m$  d'entiers.

### Exercice 3

Ecrire un programme qui lit un entier X et un tableau A du type `int` au clavier et élimine toutes les occurrences de X dans A . Le programme utilisera les pointeurs P1 et P2 pour parcourir le tableau.

Exemple : X=12

A[] = {15,12,12,11,8,10,12,11,5,12,6,12,18,20,12,16}

Résultat : A[] = {15,11,8,10,11,5,6,18,20,16}

### Exercice 4

Ecrire un programme qui demande à l'utilisateur de lui fournir un nombre entier entre 1 et 7 et qui affiche le nom du jour de la semaine ayant le numéro indiqué.

### Exercice 5

Ecrire un programme qui lit deux tableaux d'entiers A et B et leurs dimensions N et M au clavier et qui ajoute les éléments de B à la fin de A. Utiliser deux pointeurs PA et PB pour le transfer et afficher le tableau résultant A.

## Exercice 6

Ecrire de deux façons différentes, un programme qui vérifie sans utiliser une fonction de `< string >`, si une chaîne CH introduite au clavier est un palindrome :

- Q1. En utilisant uniquement le formalisme tableau
- Q2. en utilisant des pointeurs au lieu des indices numériques

## Exercice 7

- Q1. Ecrire une fonction d'entête `int EstPalindrome(char * ch)` qui retourne 1 si la chaîne de caractères ch est un palindrome ou 0 sinon.
- Q2. Ecrire une fonction d'entête `int Mystrlen(char * ch)` qui retourne la longueur de la chaîne à l'aide d'un pointeur P. La fonction n'utilisera pas de variables numériques.

## Exercice 8

- Q1. Ecrire une fonction d'entête `void printMat(int **mat, int n, int m)` qui affiche une matrice M d'ordre n x m d'entiers.
- Q2. Ecrire une fonction d'entête `float SommeMat(float **mat, int n, int m)` qui retourne la somme des éléments de la matrice mat. La fonction n'utilisera pas de variables numériques.
- Q3. Ecrire une fonction d'entête `float ProduitMat(float **mat, int n, int m)` qui retourne le produit des éléments de la matrice mat. La fonction n'utilisera pas de variables numériques.
- Q3. La transposée d'une matrice mat est une matrice dont les lignes sont les colonnes de mat. Quelles sont les dimensions de la transposée d'une matrice de taille n x m ? Ecrire la fonction `void matTransposee(int **mat, int **res, int n, int m)` qui place dans res la transposée de mat.

## Exercice 9

- Q1. Ecrire une fonction d'entête `int CharToInt(char *ch)` permettant de convertir la chaîne chiffre ch sous la forme d'un entier.  
Exemple l'appel de la fonction `CharToInt("123")` retourne 123
- Q2. Ecrire une fonction d'entête `char * IntToChar(int n, char *ch)` permettant de convertir la valeur entière n sous la forme d'une chaîne de caractères ch.  
Exemple l'appel de la fonction `IntToChar(5362)` retourne "5362"