

## PROGRAMMATION C TD2

LICENCE MATHS-INFO  
23 JANVIER 2012

### EXERCICE 1

Ecrivez une *fonction* qui, étant donnés deux tableaux de nombres entiers  $A$  et  $B$ , de même longueur  $N$ , calcule et renvoie le nombre de valeurs de  $i$  pour lesquelles on a  $A_i = B_i$ .

Ecrivez un *programme* pour tester cette *fonction*.

### EXERCICE 2

Ecrivez une *fonction* qui cherche et renvoie la *valeur* du plus petit élément d'un tableau  $T$  de  $N$  nombres entiers.

Ecrivez une *fonction* qui cherche et renvoie le *rang* du plus petit élément d'un tableau  $T$  de  $N$  nombres entiers.

Ecrivez un *programme* pour tester ces *fonctions*.

### EXERCICE 3

*Tassage*. Etant donné un tableau  $T$  de  $N$  nombres positifs ou nuls, écrivez une *fonction* qui le *tasse*, c'est-à-dire qui détecte les éléments nuls du tableau et qui récupère leur place en décalant vers le début du tableau tous les autres éléments.

### EXERCICE 4

*Recherche dichotomique dans un tableau ordonné*. On considère un tableau  $T$  de  $N$  nombres entiers deux à deux distincts, rangés par ordre croissant, et un nombre  $X$ . Ecrivez une *fonction* qui détermine l'indice exprimant soit le rang de  $X$  dans  $T$  soit, si  $X$  ne figure pas dans  $T$ , le rang de l'emplacement dans lequel il faudrait ranger  $X$  pour l'insérer dans le tableau, en conservant trié ce dernier.

Principe : considérer deux indices  $i$  et  $j$  tels que le sous-tableau  $[t_i, \dots, t_j]$  soit seul susceptible de contenir  $X$  (initialement,  $i = 0$  et  $j = N - 1$ ). En comparant  $X$  et l'élément du milieu, déterminer celle des deux moitiés du sous-tableau qui est susceptible de contenir  $X$ . Recommencer cette opération jusqu'à déterminer une unique position du tableau.

Estimez le nombre moyen d'opérations faites par ce programme.

## EXERCICE 5

*Calcul de  $P(X)$ .* Ecrivez une fonction qui calcule et renvoie la valeur d'un polynôme  $P(X)$ , représenté par le tableau de ses coefficients, pour une valeur donnée de la variable  $X$ .

## EXERCICE 6

*Puissance d'un nombre.* Ecrivez une fonction puissance qui calcule  $X^N$  pour  $X$  flottant quelconque et  $N$  entier positif ou nul. Mettez à profit la remarque suivante :

$$X^N = \begin{cases} X \times X^{N-1} & \text{si } N \text{ impair } (N - 1 \text{ est alors pair}) \\ X \times (X^2)^{\frac{N}{2}} & \text{si } N \text{ pair } (\frac{N}{2} \text{ est alors pair}) \end{cases}$$

Ecrivez un *programme* pour tester cette *fonction* en comparant le résultat obtenu avec celui de la fonction `pow` disponible dans `math.h` (cf. p. 121 du poly).

## EXERCICE 7

Ecrire un fichier `makefile` unique permettant de compiler l'ensemble des fichiers suivants :

- le fichier `compare_tableau.c` qui contient la fonction de l'exercice 1 ;
- le fichier `test_compare_tableau.c` qui contient le programme de l'exercice 1 ;
- le fichier `min_tableau.c` qui contient les fonctions de l'exercice 2 ;
- le fichier `test_min_tableau.c` qui contient le programme de l'exercice 2 ;
- le fichier `tassage.c` qui contient la fonction de l'exercice 3 ;
- le fichier `recherche_dichotomique.c` qui contient la fonction de l'exercice 4 ;
- le fichier `test_recherche_dichotomique.c` qui contient le programme de l'ex. 4 ;
- le fichier `calcule_polynome.c` qui contient la fonction de l'exercice 5 ;
- le fichier `puissance.c` qui contient la fonction de l'exercice 6 ;
- le fichier `test_puissance.c` qui contient le programme de l'exercice 6.