

# TD N°7 : Programmation

## Exercice 1

En Pascal, les fonctions trigonométriques prédéfinies sont  $\cos(x)$ ,  $\sin(x)$  et  $\arctan(x)$ . Définir la fonction  $\tan(x)=\sin(x)/\cos(x)$ , puis écrire un programme qui fait appel à cette fonction.

## Exercice 2

Soient deux vecteurs U et V ayant chacun N composantes saisies au clavier. On souhaite déterminer si ces vecteurs sont égaux. Ecrire la fonction  $\text{EqVec}(U,V)$  qui effectue la comparaison des composantes les unes après les autres et qui retourne la valeur booléenne **false** dès que deux composantes diffèrent. Au cas où les composantes de U et V sont toutes égales deux à deux, la fonction  $\text{EqVec}(U,V)$  retourne alors la valeur booléenne **true**.

## Exercice 3

Ecrire les procédures  $\text{MinTab}$  et  $\text{MaxTab}$  qui recherchent respectivement le plus petit et le plus grand élément d'un tableau. Ecrire un programme qui utilise les deux procédures précédentes pour afficher le plus petit et le plus grand élément d'un tableau. Les cases du tableau ayant été remplies auparavant par des valeurs aléatoires.

Montrez le passage de ces procédures à des fonctions qui réalisent la même chose.

## Exercice 4

Soit un tableau T de N nombres entiers. On désire tasser ce tableau en supprimant les valeurs nulles, mais en conservant le classement des autres valeurs. Pour cela, on utilise un tableau auxiliaire T\_bis dans lequel on recopie successivement toutes les valeurs non nulles de T.

- Ecrire l'en-tête du programme comportant les déclarations nécessaires ;
- Ecrire la procédure Tasser qui utilise un tableau auxiliaire T\_bis et qui a pour paramètres le tableau T et sa taille ainsi que le tableau T\_bis et sa taille ;
- Ecrire le corps du programme qui effectue la lecture de la taille du tableau, la lecture du tableau lui-même et tasse le tableau selon la méthode décrite, puis enfin affiche le tableau tassé.

→ TP 6

## Exercice 5 : nombres parfaits

Un nombre entier est qualifié de parfait si il est égal à la somme de ses diviseurs (lui-même étant exclu). Par exemple,  $28=1+2+4+7+14$  est un nombre parfait. Ecrire un programme permettant de déterminer les nombres parfaits inférieurs à un entier N saisi au clavier. Utiliser pour cela une fonction  $\text{Parfait}(i)$  qui retourne la valeur booléenne **true** lorsque i est un nombre parfait.

→ TP 6