

## Série N°4

### Les procédures et les fonctions

#### Exercice 1

1. Ecrivez une fonction *Max* qui détermine la valeur maximale de deux nombres réels passés en paramètre.
2. Ecrivez une fonction *Min* qui détermine la valeur minimale de deux nombres réels passés en paramètre.
3. Ecrivez une procédure *Echanger* qui prend deux paramètres de type réel, et ensuite réalise la permutation de leurs contenus.
4. Ecrivez l'algorithme principal qui lit au clavier deux nombres réels, et ensuite affiche la valeur minimale, la valeur maximale et le résultat de la permutation de leurs contenus.

#### Exercice 2

1. Ecrivez les trois procédures suivantes :
  - a. Une procédure *SurfaceRectangle* qui lit au clavier la longueur des arrêtes d'un rectangle et ensuite calcule et affiche sa surface;
  - b. Une procédure *SurfaceTriangle* qui lit au clavier la base et la hauteur d'un triangle et ensuite calcule et affiche sa surface;
  - c. Une procédure *SurfaceDisque* qui lit au clavier le rayon d'un disque et ensuite calcule et affiche sa surface.
2. Ecrivez une procédure *Surface* qui prend en paramètre un caractère qui désigne la surface à calculer ('r' : rectangle, 't' : triangle, 'd' : disque) et selon la valeur de celui-ci elle appelle la procédure qui calcule la surface correspondante.
3. Ecrivez l'algorithme principal qui permet à l'utilisateur de calculer et afficher selon son choix l'une des trois surfaces précédentes.

#### Exercice 3

On veut écrire un algorithme qui calcule le résultat en fonction de  $x$  et  $y$  de la fonction suivante :

$$f(x, y) = g(x) \times h(y)$$

Avec :

$$g(x) = \frac{1}{n} \sum_{i=1}^n x^i$$

$$h(y) = 1 + \frac{y}{1!} + \frac{y^2}{2!} + \dots + \frac{y^n}{n!}$$

1. Ecrivez les fonctions  $g$  et  $h$ .
2. Ecrivez la fonction  $f$ .
3. Ecrivez le programme de test.

#### Exercice 4

1. Ecrivez une procédure *LireVecteur* qui permet de lire les  $N$  éléments d'un tableau de type réel ;
2. Ecrivez une procédure *EcriresVecteur* qui permet d'afficher un tableau de  $N$  éléments de type réel ;
3. Ecrivez la fonction *ProduitVecteurs* qui calcule le produit scalaire de deux vecteurs passés en paramètre.
4. Ecrivez l'algorithme principal qui lit au clavier deux vecteurs de même taille, ensuite calcule et affiche leur produit scalaire. La mémoire est gérée dynamiquement.

#### Exercice 5

1. Ecrivez une procédure *LireMatrice* qui permet de lire les  $N \times M$  éléments d'une matrice de type réel ;
2. Ecrivez une procédure *EcrireMatrice* qui permet d'afficher une matrice de  $N \times M$  éléments de type réel ;
3. Ecrivez une fonction *AllouerMatrice* qui permet d'allouer l'espace mémoire nécessaire pour une matrice de type réel de taille  $N \times M$  ;
4. Ecrivez une procédure *LibérerMatrice* qui permet de libérer l'espace mémoire occupé par une matrice de type réel de taille  $N \times M$  ;
5. Ecrivez une fonction *TransposéeMatrice* qui calcule la transposée d'une matrice passée en paramètre;
6. Ecrivez l'algorithme principal qui lit au clavier une matrice de type réel de taille  $N \times M$  et ensuite affiche ses éléments et calcule et affiche sa transposée. La mémoire est gérée dynamiquement.