

## TP – Informatique 2

### Série de TP N°3 – Sous-Programmes : Procédures et Fonctions

**Exercice N°1 :** Soit le programme PASCAL suivant :

<pre> Program CalculPuissance; Var k: real;    { La base }     m: integer; { L'exposant }     result: real; { Le résultat de k^m } <b>Procedure</b> Calc_Puissance(<b>k: real; m: integer; Var P: real</b>);     Var i: integer;     <b>Begin</b>         <b>If</b> m = 0 <b>then</b>             P := 1 { Tout nombre élevé à la puissance 0 donne 1 }         <b>Else</b>             <b>Begin</b>                 P := 1;                 <b>For</b> i := 1 <b>to</b> m <b>do</b>                     P := P * k; { On multiplie P par k, m fois }             <b>End</b>;         <b>End</b>; <b>Procedure</b> Affich_Resultat(<b>k: real; m: integer; result: real</b>);     <b>Begin</b>         WriteLn('Le resultat de ', k:0:2, '^', m, ' est: ', result:0:2);     <b>End</b>; <b>Begin</b>     Write('Entrez la base (k) et l'exposant (m): ');     ReadLn(k,m);     Calc_Puissance(k, m, result); { Calculer la puissance }     Affich_Resultat(k, m, result); { Afficher le résultat } <b>End</b>.         </pre>	<p><b>Questions :</b></p> <ol style="list-style-type: none"> <li>1) Exécuter le programme pour k=3, m=3.</li> <li>2) Quels sont les paramètres formels et effectifs.</li> <li>3) Quels sont les paramètres à passage par valeur et ceux à passage par variable de la fonction puissance?</li> <li>4) Dérouler le programme pour k=3 et m=3.</li> <li>5) Transformer la procédure <b>Calc_Puissance</b> en une fonction <b>Calc_puissance</b> et ré-exécuter le programme pour k=3, m=3.</li> </ol>
---	--

**Exercice N°2 :**

<pre> Program Somme; <b>Procedure</b> Sous_Prog1(x, y: real; s: real);     <b>Begin</b>         s := x + y;     <b>End</b>; <b>Procedure</b> Sous_Prog2(x, y: real; var s: real);     <b>Begin</b>         s := x + y;     <b>End</b>; <b>Var</b>     a, b, c: real; <b>Begin</b>     a := 10; b := 5; c := 0;     Sous_Prog1(a, b, c);     WriteLn('La somme S-Prog1 est: ', c:6:2);     a := 10; b := 5; c := 0;     Sous_Prog2(a, b, c);     WriteLn('La somme S-Prog2: ', c:6:2); <b>End</b>.         </pre>	<p><b>Questions:</b></p> <ol style="list-style-type: none"> <li>1) Exécuter le programme.</li> <li>2) Quelle est la différence entre les deux procédures sous_prog1 et sous_prog2 ?</li> <li>3) Quels sont les paramètres à passage par valeur et ceux à passage par variable ?</li> <li>4) Quels sont les paramètres formels des deux Procédures ?</li> </ol>
	<p><b>Exercice N°3 :</b></p> <p>Soient V1 et V2 deux vecteurs de même taille contenant des nombres entiers. À l'aide de sous-programmes, on veut :</p> <ol style="list-style-type: none"> <li>1) Calculer la somme des éléments de V1 et de V2.</li> <li>2) Comparer les deux vecteurs en fonction de leurs sommes respectives.</li> </ol> <p>- Quelle est la différence entre une procédure et une fonction?</p>

**Exercice N°4 :**

En utilisant la fonction **Puissance** de l'exo1 et une autre fonction **fact** qui calcule le factoriel d'un entier n tel que  $n! = 1 \times 2 \times 3 \times \dots \times n$ ,

Écrire un programme Pascal pour calculer la valeur approximative de:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}.$$

## TP – Informatique 2

---

### Série de TP N°3 – Exercices Supplémentaires : Sous-Programmes (Procédures et Fonctions)

---

#### Exercice Sup - 01:

Écrire un programme en pascal qui calcule la somme  $e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ . Le programme devra inclure une fonction **FACT(n)** pour calculer la valeur de  $n!$ . Cette fonction doit être définie et intégrée dans le programme principal.

Enfin, affichez le résultat dans le programme principal, en veillant à ce que la valeur produite représente une approximation de **l'exponentielle e**.

#### Exercice Sup - 02 :

Écrire un programme en pascal qui permet de lire un tableau **T** de **N réels**. Le programme doit faire appel à une procédure qui détermine le plus grand élément du tableau **T** ainsi que sa position. Ensuite, les résultats doivent être affichés dans le programme principal.

#### Exercice Sup – 03 :

1. **Créer une procédure Permuter** : Écrivez une procédure qui prend en entrée deux pointeurs de réels **X** et **Y**, et effectue la permutation de leurs valeurs.
2. **Écrire un programme en pascal qui** :
  - Lit une matrice carrée de réels de taille  $N \times N$  (où  $N$  est un entier positif).
  - Utilise la **procédure Permuter** pour échanger les éléments des deux diagonales de la matrice :
  - ✓ La première diagonale (*diagonale principale*) est constituée des éléments **T[i][i]** pour  $i=0,1,\dots,N-1$ .
  - ✓ La seconde diagonale (*diagonale secondaire*) est constituée des éléments **T[i][N-1-i]** pour  $i=0,1,\dots,N-1$ .

➤ Affiche la matrice avant et après la permutation des éléments des deux diagonales.

#### Exercice Sup – 04 :

Écrire un programme en Pascal qui calcule une approximation de **ln(1+x)** en utilisant la **série de Taylor**.

1. Écrire une **fonction Puissance** qui calcule  $a^n$  pour un nombre réel  $a$  et un entier  $n$ .
2. Écrire une **fonction fact** qui calcule le factoriel d'un entier  $n$ .
3. Écrire une **fonction ln\_approximation** qui calcule une approximation de **ln(1+x)** pour un nombre réel  $x$  (où  $-1 < x \leq 1$ ) et un entier  $n$ , en utilisant la série suivante :

$$\ln(1+x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + \frac{(-1)^{n-1} \cdot x^n}{n}$$

4. Dans le programme principal, demander à l'utilisateur de saisir un nombre réel  $x$  (où  $-1 < x \leq 1$ ) et un entier  $n$ , et afficher l'approximation de **ln(1+x)**.