

TP n° 4 : Modularité**Les procédures en pascal**

```
PROGRAM < nom du programme > ;  
VAR ..... ; {Déclaration des variables de programme principal (variables globales)}  
PROCEDURE < nom de la procédure > (< déclaration des paramètres formels > ) ;  
VAR ..... ; {Déclaration des variables de la procédure (variables locales)}  
BEGIN {Début de code source de la procédure}  
< Bloc d'instructions >  
Writeln('Le résultat est :', < résultat de la procédure > ;  
END; {Fin de code source de la procédure}  
  
BEGIN {Début du programme principal}  
Writeln('Introduire .....') ;  
Readln(< variables d'entrées > ) ; {Lecture des variables d'entrées}  
< nom de la procédure > (< paramètres réels > ) ; {Appel à de la procédure}  
END. {Fin du programme principal}
```

- Le mot clé **PROCEDURE** débute une procedure. Après ce mot vient le nom de la procédure et ses éventuels paramètres.
- Les variables du programme principal sont globales.
- Pour passer un paramètre par adresse, il faut précéder son nom par le mot clé **VAR**. Si **var** est absent, le passage est réalisé par valeur.

Les fonctions en pascal

```
PROGRAM < nom du programme > ;  
VAR ..... ; {Déclaration des variables de programme principal (variables globales)}  
FUNCTION < nom de la fonction > (< déclaration des paramètres formels > ) : < Type du résultat > ;  
VAR ..... ; {Déclaration des variables de la fonction (variables locales)}  
BEGIN {Début de code source de la fonction}  
< Bloc d'instructions >  
< nom de la fonction > := < résultat de la fonction > ;  
END; {Fin de code source de la fonction}  
  
BEGIN {Début du programme principal}  
Writeln('Introduire .....') ;  
Readln(< variables d'entrées > ) ; {Lecture des variables d'entrées}  
< variable de sortie > := < nom de la fonction > (< paramètres réels > ) ; {Appel à de la fonction}  
Writeln('Le résultat est :', < variable de sortie > ) ;  
END. {Fin du programme principal}
```

- Le mot clé **FUNCTION** débute une fonction. Après ce mot vient le nom de la fonction, ses éventuels paramètres et le type de la valeur retournée.
- On retourne un résultat d'une fonction en écrivant une affectation sur son nom (<nom fonction>:= <résultat de la fonction>).

Exercice 1 :

Écrire un programme utilisant une fonction ou une procédure '**Affiche-Pairs**' qui permet de lire une liste de nombres entiers dont la dernière valeur = -1 et qui affiche le nombre d'entiers pairs. Le programme utilisera ensuite une fonction ou une procédure '**Pourcentage_Pairs**' permettant de calculer le pourcentage des nombres pairs par rapport au nombre d'entiers donnés.

Exercice 2 : Exponentiel

Écrire un programme permettant de calculer e^x selon son développement limité :

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

Exercice 3: Phrase

Soit une phrase qui se termine par un point. Les mots de cette phrase sont séparés par un ou plusieurs espaces. Ecrire un programme qui fait appel à une procédure pour :

- Déterminer le nombre de mots de cette phrase.
- Afficher le mot de la taille maximale (le plus long).

Exercice 4: Partie entière et partie fractionnaire

En utilisant des fonctions /procédures, écrire un programme qui permet de calculer la partie entière et la partie fractionnaire d'un nombre positif.

Exercice 5 : Espace de stockage d'un disque dur

Dans un PC, la taille globale de son disque dur est de 120 GO. On veut le partitionner en 3 partitions : C, D et E, tel que :

- C : Représente la moitié de l'espace global.
- D : Représente les 2 tiers de l'espace restant.
- E : Représente l'espace restant.

On alloue pour la corbeille un espace de 15% de la partition C, 10% de la partition D et 10% de la partition E.

- Calculer l'espace de chaque partition (utiliser une fonction ou une procédure).
- Quel est l'espace réel de stockage pour ce disque dur ?

Exercice 6 : Mot palindrome

En utilisant des fonctions/procédures, écrire un programme pascal qui vérifie si un mot donné par l'utilisateur est un palindrome. Un palindrome est un mot qui se lit de la même façon de gauche à droite ou de droite à gauche (**exemple** : ICI, RADAR, ...).