LES PROCEDURES ET LES FONCTIONS

INTRODUCTION

En général, les problèmes qui se posent en pratique sont suffisamment complexes et nécessitent leurs décompositions en sous problèmes plus faciles à résoudre séparément. De là vient l'idée de décomposition d'un programme complexe en sous-programmes (***fonctions et procédures***) plus facile à écrire et à contrôler. Ces fonctions et procédures une fois rassemblées, peuvent communiquer des résultats entre elles et avec le programme principal.
Une autre raison d'usage des fonctions et procédures est pour éviter des redondances de codes dans le programme.

NOTION DU SOUS-PROGRAMME

Un sous-programme (***procédure ou fonction***) est un programme à l'intérieur d'un autre programme' Il possède la même structure que le programme principal. I1 peut être appelé par le programme principal ou par un autre sous-programme pour réaliser un certain traitement et leur retourner des résultats (un ou plusieurs résultats).

*La différence principale entre **les procédures** et **les fonctions** est que **les procédures** ne possède pas de valeurs (elles ne retournent pas directement de résultats). Par contre, **les fonctions** retournent directement une valeur (un résultat). Cependant, toute procédure peut être convertie en une **fonction** et vice-versa (toute **fonction** peut être convertie en **procédure**).*

LES PROCÉDURES

On désigne par ***procédure*** une séquence d'instructions de programme à laquelle on associe un nom spécifique représenté par un identificateur. On distingue deux catégories de procédures : les procédures standards et ***les procédures*** non standards.

- Par ***procédure standard***, on entend toute procédure connue du langage, donc qui se passe d'une déclaration préalable lors de son utilisation dans un programme. C'est le cas des instructions de lecture et d'écriture **READ** et **WRITE** et de leurs variantes **READLN** et **WRITELN**.
- Par ***procédure non standard***, on entend toute procédure non connue du langage et qui, par conséquent, nécessite une définition préalable à son utilisation. C'est précisément cette catégorie de procédures qui retient notre attention dans ce chapitre.

La structure générale d'un programme PASCAL comportant des fonctions et des procédures est comme suit:

Program <Nom_programme>;
uses wincrt;
<Déclaration des constantes éventuelles>;
<Déclaration des types éventuels>;
<Déclaration des variables éventuelles>;
<Déclaration des fonctions>;
<Déclarations des procédures>;
BEGIN
<Instruction 1> ;
<Instruction 2>;
……
……
<Instruction N>;
END.

Ainsi, les sous-programmes (procédures et fonctions) sont déclarés dans la partie déclaration du programme principal avec les constantes, variables, types, etc.

DECLARATION D'UNE PROCEDURE

Une procédure possède la même structure que le programme principal, pour définir une procédure, on suit le modèle suivant :

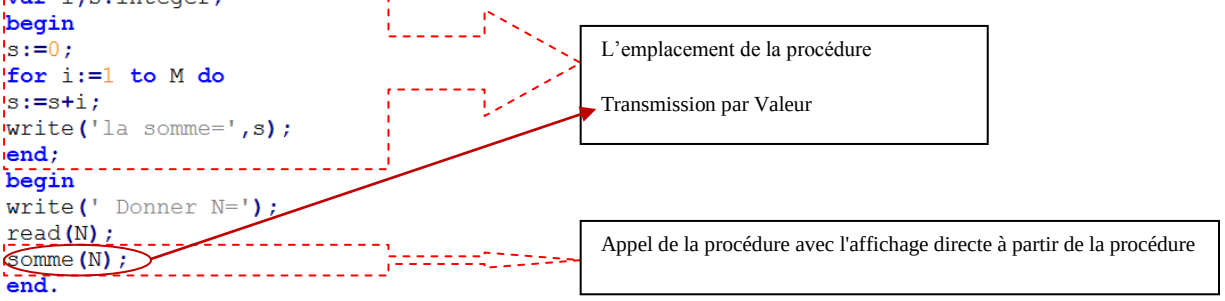
1- Transmission par valeur :

Algorithme	Pascal
Procédure Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN) Déclarations des variables et les constants ; **Début** Instructions ; …… Instructions ; **Fin.**	**Procedure** Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN) ; Déclarations des variables et les constants ; **Begin** Instructions ; …… Instructions ; **End.**

<Nom-Procdure> : identificateur de la procédure (il faut respecter les règles des identificateurs).
<paraml>, <pran2> ... <paramN> : sont des paramètre de la procédure.
<type1>/ <type2> ... <typeN> : sont les types respectifs des paramètres de laprocédure.
Comme le programme principal, une procédure possède une partie déclaration et une partie instructions qui délimitée par Begin etEnd

Exemple :
En utilisant une procédure calcules cette somme : S=1+2+3+4+…..+N ;

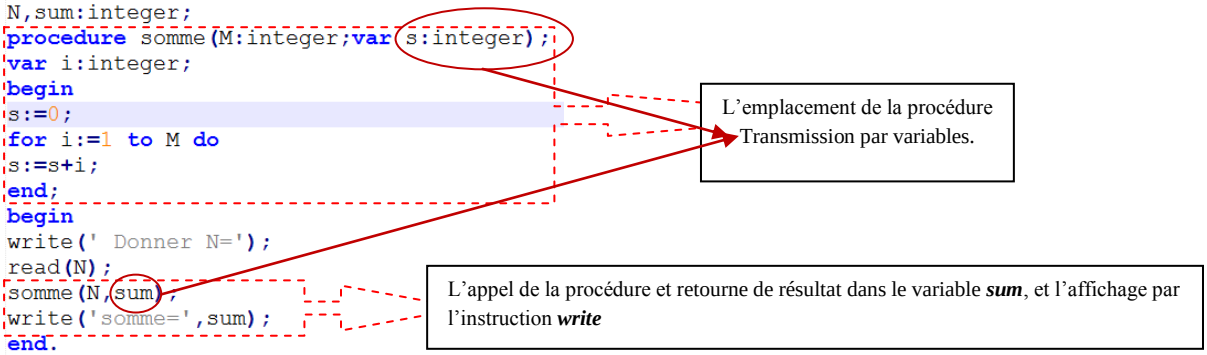
```
Program exo1_a;
Uses wincrt ;
var
N   :integer;
procedure somme(M:integer);
var i,s:integer;
begin
s:=0;
for i:=1 to M do
s:=s+i;
write('la somme=',s);
end;
begin
write(' Donner N=');
read(N);
somme(N);
end.
```



2- Transmission par variables

Algorithme	Pascal
Procédure Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN ; Var : varia1 :type 1; vari2 :type2….. ;) ; Déclarations des variables et les constants ; **Début** Instructions ; …… Instructions ; **Fin.**	**Procedure** Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN ; Var varia1 :type1 ; vari2 :type2….. ;) ; Déclarations des variables et les constants ; **Begin** Instructions ; …… Instructions ; **End.**

```
Program Somm;
Uses wincrt ;
var
N,sum:integer;
procedure somme(M:integer;var s:integer);
var i:integer;
begin
s:=0;
for i:=1 to M do
s:=s+i;
end;
begin
write(' Donner N=');
read(N);
somme(N,sum);
write('somme=',sum);
end.
```



DECLARATION D'UNE FONCTION

Comme les procédures, une fonction possède la même structure que le programme principal, à la différence, elle possède un type de retour (de résultat). Pour définir une fonction, on suit le modèle suivant :

Algorithme	Pascal
fonction Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN) : Type_fonction ; Déclarations des variables et les constants ; **Début** Instructions ; …… Instructions ; **Fin.**	**Fuction** Nom_ident(param1 :type1 ;param2 :type1 ;… ;paramN :typeN) : Type_function ; Déclarations des variables et les constants ; **Begin** Instructions ; …… Instructions ; **End.**

Exemple (refaire l'exemple précédent) :
Remplacer la procédure par une fonction.

```
Program fonction_somme;
Uses wincrt ;
var
N,sum:integer;
function somme(M:integer):integer;
var i,s:integer;
begin
s:=0;
for i:=1 to M do
s:=s+i;
somme:=s;
end;
begin
write(' Donner N=');
read(N);
sum:=somme(N);
write('somme=',sum);
end.
```

L'emplacement de la fonction.
Définition de type de la fonction.
Le résultat retourne dans le nom de la fonction (dans ce cas : ***somme***)
A la fin de chaque fonction, on doit affecte le résultat à la fonction, ***cette opération est obligatoire***.

On peut exploite le résultat sans l'affecte à une autre variable.
On peut afficher le résultat directement
write ('somme=', somme(N)) ;

TRANSMISSION DES PARAMETRES PAR FONCTION

Écrire un programme pascal qui calcule la somme suivante :
Somme=$1^1+2^2+3^3+4^4+\ldots\ldots+N^N$ ***Remarque (Transmission des paramètres par fonction)***

```
program transm_para_fonction;
uses wincrt;
var N,sum:integer ;
function puis(p:integer):integer;
var i,fc:integer;
begin
fc:=1;
if p>1 then
for i:=1 to p do
fc:=p*fc;
puis:=fc;
end;
procedure somme (k:integer; var sum:integer);
var j:integer;
begin
sum:=0;
for j:=1 to k do
sum:=sum+puis(j);
end;
begin
write('donnez N=');
read (N);
somme(N,sum);
write('somme=', sum)
end.
```

L'emplacement de la fonction

-L'emplacement de la procédure
-Appel de la fonction à partir de la procédure
(transmission des paramètres par fonction

Exo2 : même question pour cette somme S=1 !+2 !+3 !+…….+N !

```
program procedure1;
uses wincrt;
var N,sum:integer ;
function fact(p:integer):integer;
var i,fc:integer;
begin
fc:=1;
for i:=1 to p do
fc:=fc*i;
fact:=fc;
end;
procedure somme (k:integer; var sum:integer);
var j:integer;
begin
sum:=0;
for j:=1 to k do
sum:=sum+fact(j);
end;
begin
write('donnez N=');
read (N);
somme(N,sum);
write('somme=', s)
end.
```

L'emplacement de la fonction

-L'emplacement de la procédure
-Appel de la fonction à partir de la procédure
(transmission des paramètres par fonction