

fonctions

exemple : définition itérative de $x \rightarrow x^n, n \in \mathbb{N}$:

```
function puiss(x:real,n:integer):real;
var i: integer;xpn:real;
begin
  xpn:= 1;
  for i:= 1 to n do xpn:= xpn*x;
  puiss:= xpn;
end;
```

commentaires :

déclaration de la fonction ; x et n sont les paramètres de la fonction
déclaration des variables locales
l'appel `write(puiss(3,-2))` de cette fonction donnerait un résultat faux ($-2 < 0$);
la variable intermédiaire `xpn` est indispensable

exemple : définition récursive de $x \rightarrow x^n, n \in \mathbb{N}$:

```
function puiss(x:real,n:integer):real;
begin
  if n=0 then puiss:= 1
  else puiss:=x*puiss(x,n-1);
end;
```

commentaires :

appel récursif de la fonction

procédures

exemple : tri de deux nombres: si $a > b$, on échange a et b...

```
program tri;
var a,b:real;
procedure echanger(var x,y:real):
  var aux:real;
  begin
    aux:=x;x:=y;y:=aux;
    writeln(x,y);
  end;
BEGIN
  write('a=');readln(a);
  write('b=');readln(b);
  if a>b then echanger(a,b);
  writeln(a,b);
END.
```

variante :

```
procedure echanger(x,y:real):
```

Exécution : a = 5 ↴ b = 3 ↴

Résultats :

3	5	écriture dans le corps de la procédure	la variante donne:	3	5
3	5	écriture dans le programme principal		5	3

commentaires :

a, b sont les variables du programme principal ; x, y sont les paramètres de la procédure ; **var** indique que le passage des paramètres dans le programme principal se fait **par variable**, ce qui veut dire que le contenu des variables du programme principal est affecté par l'appel de la procédure . Dans la variante, où le **var** est omis, le contenu des variables du programme principal n'est pas affecté par l'appel de la procédure (passage des paramètres **par valeur**).

Règle : On transmet les paramètres qui correspondent à des variables d'entrée par valeur, à des variables de sortie par variable.