TP Informatique 2

Corrigé de la série de TP N°3 – Sous programmes : Fonctions et Procédures

Exercice N°01 :

Soit le programme pascal suivant :

```
Program Exo_1;
Var x, y : integer;                          {variables globales}
Procedure Proc1(A: integer; Var B: integer) ;    {SP1}
Begin
    A := A + 1;  B := 22;
End;                                         {Fin SP1}
Procedure Proc2(A: integer; B : integer) ;       {SP2}
Begin
    A := A + 1;  B := 22;
End;                                         {Fin SP2}
Begin                                        {PP}
  x := 3;  y := 7;
  Proc1(x, y);                               {Appel au SP1}
  Writeln('x= ', x, ' y = ', y);
  x := 3;  y := 7;
  Proc2 (x, y);                              {Appel au SP2}
  Writeln('x= ', x, ' y = ', y);
End.                                         {Fin PP}
```

Questions :

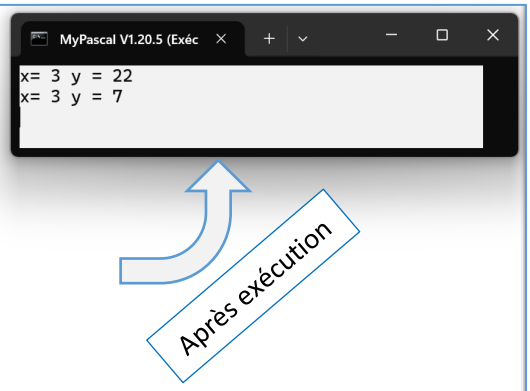
1) Exécuter le programme.
2) Quelle est la différence entre les deux procédures Proc1 et Proc2 ?
3) Quels sont les paramètres à passage par valeur et ceux à passage par variable ?
4) Quels sont les paramètres formels des deux procédures ? les paramètres effectifs ?
5) Les appels Proc1(3,7) et Proc2(3,7) sont-ils corrects, expliquer pourquoi.
6) Dérouler le programme.
7) Exécuter le programme en donnant le type « réel » à la variable y. Que se passe-t-il ? Pourquoi ?

Remarque :
On définit les acronymes SP et PP comme suit :
SP : **S**ous-**P**rogramme,
PP : **P**rogramme **P**rincipal

Solution :

1) Exécuter le programme :

```
1  Program Exo_1;
2  Var x, y : integer;                    {variables globales}
3  Procedure Proc1(A: integer; Var B: integer) ;   {SP1}
4  Begin
5      A := A + 1;  B := 22;
6  End;                                   {Fin SP1}
7  Procedure Proc2(A: integer; B : integer) ;      {SP2}
8  Begin
9      A := A + 1;  B := 22;
10 End;                                   {Fin SP2}
11 Begin                                  {PP}
12   x := 3;  y := 7;
13   Proc1(x, y);                         {Appel au SP1}
14   Writeln('x= ', x, ' y = ', y);
15   x := 3;  y := 7;
16   Proc2 (x, y);                        {Appel au SP2}
17   Writeln('x= ', x, ' y = ', y);
18 End.                                   {Fin PP}
```



2) Quelle est la différence entre les deux procédures Proc1 et Proc2?

La différence est la présence du mot clé « **var** » dans Proc1 mais pas dans Proc2.

3) Quels sont les paramètres à passage par valeur et ceux à passage par variable ?

Les paramètres à passage par valeur et ceux à passage par variable :

Paramètre/Procédure	Proc1	Proc2
Passage de paramètre par valeur	A	A et B
Passage de paramètre par variable	B	/

4) Quels sont les paramètres formels des deux procédures ?

Les paramètres formels des deux procédures :

Type de paramètres/Procédure	Proc1	Proc2
Paramètres formels	A et B	A et B

Quels sont les paramètres effectifs ?

Les paramètres effectifs des deux procédures :

Type de paramètres/Procédure	Proc1	Proc2
Paramètres effectifs	x et y	x et y

Rappel :

- Un paramètre à passage par variable (ou par adresse) est un paramètre précédé par le mot clé «**var** ».
- Les paramètres formels sont les paramètres utilisés dans la déclaration des procédures et fonctions. Par contre, les paramètres effectifs sont les paramètres utilisés lors de l'appel aux procédures et fonctions. (Les paramètres formels sont séparés par des points-virgules).

5) Les appels Proc1(3,7) et Proc2(3,7) sont-ils corrects, expliquer pourquoi ?

- Pour Proc1(3,7) : L'appel est incorrect car le paramètre formel B est en entrée / Sortie : il doit lui correspondre une variable, et non pas une constante.

- Pour Proc2(3,7) : L'appel est correct car les paramètres formels A et B sont des paramètres d'entrée seulement. Par conséquent, ils peuvent leur correspondre des variables ou des constantes.

6) Dérouler le programme.

Instructions	Variables programme principal		Variables procédure Proc1		Variables procédure Proc2		Affichage
	x	y	A	B	A	B	
x:=3	3	/					
y :=7	3	7					
Proc1(x, y) *Appel à la procédure Proc1 et transmission des paramètres*	3	7	3	7			
A := A + 1 A := 3 + 1 A := 4			4	7			
B := 22 *Proc1 a calculé la valeur de A et B, la valeur de B est retournée à la variable globale y* *(car B est passé par variable)*				22			
		22					
Writeln('x = ', x, ' y = ', y);	3	22	4	22			x = 3 y = 22
x:=3	3	22					
y :=7	3	7					
Proc2(x, y) *Appel à la procédure Proc2 et transmission des paramètres*	3	7			3	7	
A := A + 1 A := 3 + 1 A := 4					4	7	
B := 22 *Proc2 a calculé la valeur de A et B, mais contrairement à Proc1, la valeur de B n'est pas retournée à la variable globale y* *(car B est passé par valeur)*						22	
Writeln('x = ', x, ' y = ', y);	3	7			4	22	x = 3 y = 7

7) Exécuter le programme en donnant le type « réel » à la variable y. Que se passe-t-il ? Pourquoi ?

Nous obtenons **une erreur de compilation** indiquant que « un appel avec paramètre par variable doivent être de type exact de la déclaration ». Ceci revient à l'incompatibilité de types entre le paramètre effectif « y » et son paramètre formel correspondant « B ».

Remarque :

Afin d'effectuer la transmission des paramètres effectifs vers les paramètres formels, la correspondance de type et de nombre entre ces paramètres est **obligatoire.** i.e,

- Le nombre de paramètres effectifs doit être égal au nombre de paramètres formels.
- Le paramètre effectif et le paramètre formel correspondant doivent avoir le même type.

Exercice N°02 :

Soit le programme pascal suivant :

```
Program  exo_2;
Var   N: integer; Fact: integer;      {Variables globales}
  Function  Factoriel (m: integer): integer;        {SP}
  Var j, f: integer;                {Variables locales au SP}
  Begin
    f := 1;
    For j := 1 to m do
      f := f*j;
    Factoriel := f;                 {Résultat retourné}
  End;                                              {Fin SP}
Begin                                               {Début PP}
  Write('Introduire N : ');
  Read(N);
  Fact := Factoriel(N);                 {Appel au SP}
  Write('Le factoriel de ', N,  ' = ', Fact);
End.                                                {Fin PP}
```

Questions :

1) Exécuter le programme pour N = 6.
2) Réécrire le programme pour calculer la somme S suivante :

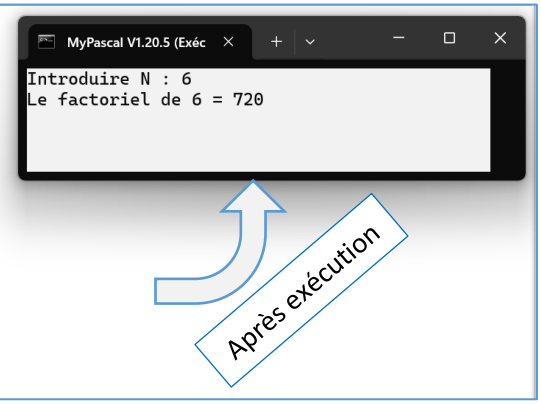
$$S = \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{N!} \qquad N \geq 1$$

3) Exécuter le programme pour N = 3.
4) Dérouler le programme pour N = 3.
5) Réécrire le programme modifié en remplaçant la fonction par une procédure de même nom.

Solution :

1) Exécuter le programme pour N = 6.

```
1   Program  exo_2;
2   Var   N: integer; Fact: integer;      {Variables globales}
3     Function  Factoriel (m: integer): integer;        {SP}
4     Var j, f: integer;                {Variables locales au SP}
5     Begin
6       f := 1;
7       For j := 1 to m do
8         f := f*j;
9       Factoriel := f;                 {Résultat retourné}
10    End;                                              {Fin SP}
11  Begin                                               {Début PP}
12    Write('Introduire N : ');
13    Read(N);
14    Fact := Factoriel(N);                 {Appel au SP}
15    Write('Le factoriel de ', N,  ' = ', Fact);
16  End.                                                {Fin PP}
```



2) Réécrire le programme pour calculer la somme S suivante :

$$S = \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{N!} \qquad N \geq 1 \quad \rightarrow \quad \text{Le terme général est : } S = \sum_{i=1}^{N} \frac{1}{i!}$$

Le programme Pascal – Solution 1	Le programme Pascal – Solution 2

Program exo_2_Q2;
Var i,N: integer; Fact: integer; S: real;
Function Factoriel (m: integer): integer;
Var j, f: integer;
Begin
f := 1;
For j := 1 to m do
f := f*j;
Factoriel := f;
End;
Begin
Write('Introduire N : ');
Read(N);
S := 0;
For i := 1 to N do
Begin
Fact := ***Factoriel(i)***;
S := S + 1/Fact;
End;
Write(' La somme S = ', S:3:1);
End.
	Program exo_2_Q2;
	Var i,N: integer; S: real;
	Function Factoriel (m: integer): integer;
	Var j, f: integer;
	Begin
	f := 1;
	For j := 1 to m do
	f := f*j;
	Factoriel := f;
	End;
	Begin
	Write('Introduire N : ');
	Read(N);
	S := 0;
	For i := 1 to N do
	Begin
	S := S + 1/Factoriel(i);
	End;
	Write(' La somme S = ', S:3:1);
	End.

3) Exécuter le programme pour N = 3

4) Dérouler le programme pour N = 3.

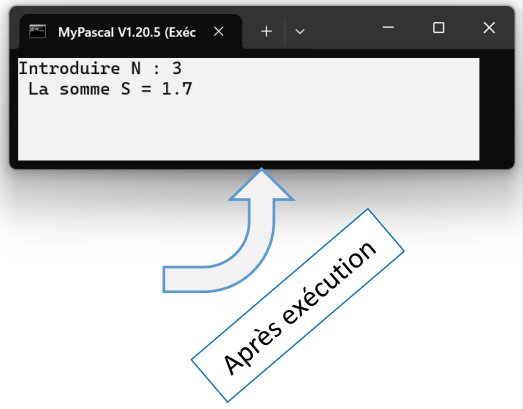
Instructions	Programme principal				La fonction Factoriel				Affichage
	N	i	Fact	S	m	j	f	Factoriel	
Write('Introduire N : ');	/	/	/	/					Introduire N :
Read(N)	3	/	/	/					
S := 0	3	/	/	0					
For i :=1	3	1	/	0					
Fact:=Factoriel(i) Fact := Factoriel(1) *(l'appel à Factoriel avec le paramètre i=1)*	3	1	**Fact := ?**	0					
=> La transmission des paramètres f:=1 for j:=1 f:=f*j → f :=1*1 = 1 Factoriel := f → factoriel := 1 *=> Le retour du résultat dans le PP*					**1**	1	1 1	**1**	
Fact := Factoriel(1) Fact := 1	3	1	1	**0**					
S := S + 1/Fact S := 0 + 1/1 S := 1	3	1	1	**1**					
For i := 2	3	2	1	1					
Fact:=Factoriel(i) Fact:=Factoriel(2) *(l'appel à Factoriel avec le paramètre i=2)*	3	2	**Fact := ?**	1					
=> La transmission des paramètres f:=1 for j:=1 f:=f*j → f := 1*1 = 1 for j:=2 f:=f*j → f := 1*2 = 2 Factoriel := f → factoriel := 2 *=> Le retour du résultat dans le PP*					**2**	1 2	1 1 2	**2**	
Fact := Factoriel(2) Fact := 2	3	2	2	**1**					
S := S + 1/Fact → S := 1 + 1/2 S := 1.5	3	2	2	**1.5**					
For i :=3	3	3	2	**1.5**					
Fact:=Factoriel(i) Fact := Factoriel(3) *(l'appel à Factoriel avec le paramètre i=3)*	3	3	**Fact := ?**	**1.5**					
=> La transmission des paramètres f:=1 for j:=1 f:=f*j → f := 1*1 = 1 for j:=2 f:=f*j → f := 1*2 = 2 for j:=3 f:=f*j → f := 2*3 = 6 Factoriel := f → factoriel := 6 *=> Le retour du résultat dans le PP*					**3**	1 2 3	1 1 2 6	**6**	
Fact := Factoriel(3) Fact := 6	3	3	6	**1.5**					
S := S + 1/Fact → S := 1.5 + 1/6 S := 1.666666666666667	3	3	6	**1.6666 66666 66667**					
Write('La somme S = ',S:3:1);	3	3	6	≠	3	3	6	6	La somme S = 1.7

5) Réécrire le programme modifié en remplaçant la fonction par une procédure de même nom.

Le programme avec fonction	Le programme avec procédure

Program exo_2_Q5;
Var i,N: integer; Fact: integer; S: real;
{Début de la fonction}
Function Factoriel (m: integer)~~: integer~~;
Var j,~~ f~~: integer;
Begin
f := 1;
For j := 1 to m do
f := f*j;
~~Factoriel := f;~~
End; **{Fin de la fonction}**
 	
Begin
Write('Introduire N : ');
Read(N);
S := 0;
For i := 1 to N do
Begin
~~Fact :=~~ Factoriel(i);
{L'appel à la fonction et affectation du résultat retourné à la variable globale Fact}
S := S + 1/Fact;
End;
Write(' La somme S = ', S:3:1);
End.
	Program exo_2_Q5;
	Var i,N: integer; Fact: integer; S: real;
	{Début de la procédure}
	Procedure Factoriel (m: integer; **Var f: integer**);
	Var j : integer;
	Begin
	f := 1;
	For j := 1 to m do
	f := f*j;
	
	End; **{Fin de la procédure}**
	Begin
	Write('Introduire N : ');
	Read(N);
	S := 0;
	For i := 1 to N do
	Begin
	Factoriel(i, **Fact**);
	{L'appel à la procedure et récuperation du résultat via le paramétre effectif Fact correspondant au paramétre formel f passé par variable}
	S := S + 1/Fact;
	End;
	Write(' La somme S = ', S:3:1);
	End.

```
1  Program exo_2_Q5;
2  Var i,N: integer; Fact: integer; S: real;
3  {Début de la procédure}
4   Procedure Factoriel (m: integer; Var f: integer);
5   Var j : integer;
6   Begin
7     f := 1;
8     For j := 1 to m do
9       f := f*j;
10
11  End;    {Fin de la procédure}
12 Begin
13  Write('Introduire N : ');
14  Read(N);
15  S := 0;
16  For i := 1 to N do
17   Begin
18     Factoriel(i, Fact);
19 {L'appel à la procedure et récuperation du résultat via le paramétre effectif Fact correspondant au paramétre formel f passé par variable}
20     S := S + 1/Fact;
21    End;
22  Write(' La somme S = ', S:3:1);
23 End.
```



Exercice N°03 :

Un nombre parfait est un nombre entier positif supérieur à 1 qui est égal à la somme de ses diviseurs excepté lui-même (on ne compte pas comme diviseur le nombre lui-même).

1) Écrire une fonction **« Parfait »** qui vérifie si un nombre entier A est parfait ou non.
2) En utilisant la fonction, écrire un programme Pascal pour chercher les nombres parfaits entre 1 et 100.

Exemple : 6 et 28 sont des nombres parfaits car : 6 = 1 + 2 + 3, 28 = 1 + 2 + 4 + 7 + 14

Solution :

Le programme Pascal – Solution 1

```
Program  exo_3;
Var  i: integer;

  Function  Parfait(A: integer): boolean ;
  Var j, S: integer;
  Begin
    S:=0;
    For j:=1 to (A div 2) do
        If (A mod j = 0) then
            S := S + j;
    If (S = A) then
      Parfait := True
    Else
       Parfait := False;
  End;

Begin
  For i := 1 to 100 do
      If Parfait(i) then
          Writeln(i, ' est un nombre parfait ');
End.
```

Le programme Pascal – Solution 2

```
Program  exo_3;
Var  i: integer;

  Function  Parfait(A: integer): boolean ;
  Var j, S: integer; P: boolean;
  Begin
    S:=0;
    For j:=1 to (A div 2) do
        If (A mod j = 0) then
            S := S + j;
    If (S = A) then
      P:= True
    Else
       P:= False;

    Parfait := P;
  End;

Begin
  For i := 1 to 100 do
      If Parfait(i) then
          Writeln(i, ' est un nombre parfait ');
End.
```

Université A. Mira de Bejaia
Faculté de Technologie
Département de Technologie

Année universitaire 2023/2024
1ère Année Technologie
Semestre 2

TP Informatique 2

Série de TP N°3 – Sous programmes : Fonctions et Procédures

Exercices supplémentaires

Exercice 01-Sup :

Écrire un programme Pascal qui permet de vérifier, à l'aide de sous-programmes « **oppose** » et « **inverse** », si deux nombres sont opposés ou non et s'ils sont inverses ou non, respectivement.

Deux nombres sont opposés si leur somme est égale à 0.

Deux nombres sont inverses si leur produit est égal à 1.

Exercice 02-Sup :

1) Écrire une fonction « **Puiss** » qui calcule la puissance énième de X (X^N) :

$$\text{Puiss}(X,N) = \begin{cases} 0 & \text{Si } X = 0 \\ 1 & \text{Si } N = 0 \\ X \times X \times \ldots \times X \text{ (N fois)} & \text{Si } N > 1 \end{cases}$$

2) En utilisant la fonction « Puiss », écrire un programme Pascal qui permet de calculer la somme S :

$$S = X^1 + X^3 + X^5 + \ldots. + X^{2N+1}$$

Exercice 03-Sup :

1) Écrire une procédure « **occurences** » qui calcule le nombre d'occurrences d'un chiffre ($0 \leq$ chiffre ≤ 9) dans un tableau T de N éléments entiers.

Exemples:

Le nombre d'occurrences du chiffre 7 dans T = | 7 | 7 | 8 | est 2.

Le nombre d'occurrences du chiffre 1 dans T = | 7 | 7 | 8 | est 0.

2) Écrire la procédure, l'insérer dans le programme et afficher les résultats dans le programme principal.

Exercice 04-Sup :

1) Écrire une procédure « **Max_vecteur**» qui calcule le plus grand élément du tableau T de K composantes réelles.
2) En utilisant la procédure « **Max_vecteur**», écrire un programme Pascal qui permet de calculer et afficher le plus grand élément de chaque colonne d'une matrice M carrée d'ordre N.

Exercice 05-Sup :

1) Écrire une procédure « **Décaler** » qui permet de décaler de manière circulaire à droite les éléments d'un vecteur T de M composantes entières.
2) En utilisant la procédure « **Décaler** », écrire un programme Pascal qui permet de lire les éléments d'une matrice A de taille N×M, décaler de manière circulaire à droite les éléments de chaque ligne et afficher la matrice résultat.