

S é r i e de TP N° 2 d'Informatique II (fonctions et Procédures) (Corrigé)

Exercice 1 : Exécuter le programme suivant :

```
1 Program Exo_1;
2 Uses wincrt ;
3 Var a, b, c : real;{Variables Globales du programme}
4 Procedure Proc1(x:real ; y:real ; var s:real) ; {sous-programme procedure n° 1}
5 Begin
6   s := x/y ;
7 End;
8 Procedure Proc2 (x:real ; y:real ; s:real); {sous-programme procedure n° 2}
9 Begin
10  s := x/y;
11 End;
12 BEGIN {Début du corps Programme Principal}
13  a:=10; b:=5; c:=0;
14  Proc1(a, b, c);
15  Writeln('Le résultat est : ', c);
16  a:=10; b:=5; c:=0;
17  Proc2 (a, b, c);
18  Writeln('Le résultat est : ', c);
19 End. {Fin du Programme Principal}
```

Après exécution du programme nous obtenons :

Le résultat est : 2

Le résultat est : 0

- 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.

- Quels sont les paramètres à passage par valeur et ceux à passage par variable ? Les paramètres à passage par valeur sont : x, y de Proc1 et x, y, s de Proc2

Les paramètres à passage par variable (ou par adresse) : s de Proc1 (le paramètre précédé par le mot clé **var**)

- Quels sont les paramètres formels des deux procédures ? Les paramètres formels sont : x, y, s de Proc1 et Proc2

- Et quels sont les paramètres effectifs ? Les paramètres effectifs sont : a, b, c de Proc1 et Proc2.

- Exécuter le programme en donnant le type integer à la variable c. que se passe-t-il? et pourquoi? Nous obtenons une erreur de compilation (*type mismatch*).

Note :

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 = nombre de paramètres formels.

- Le paramètre effectif et le paramètre formel correspondant doivent avoir le même type (ou de type compatible).

Exercice 2 : Exécuter le programme suivant :

```

1 Program CombinaisonNK;
2 Uses wincrt ;
3 Var n, k, c : integer; {Variables Globales du programme}
4 Function fact (n:integer):integer ;
5 Var f, i : integer ; {Variables locales de la fonction fact}
6 Begin
7   F:=1;
8   For i:=1 to n do
9     f:=f*i;
10    fact:=f; {Une fonction se termine toujours par une affectation}
11  End;
12 BEGIN {Début du Programme Principal}
13 Writeln ('Donnez la valeur de n et k :');
14 Read (n, k);
15 C:= fact(n) div ( fact(k) * fact(n-k) );
16 Writeln ('La combinaison de k à partir de n = ', c);
17 End. {Fin du Programme Principal}

```

- Dérouler le programme ci-dessus pour n = 5 et k = 3

Instructions	Variables du programme principal			Variables de la fonction			
	n	k	c	n	f	i	fact
Read (n,k)	5	3					
Fact(k)			3		1		
					1	1	
					2	2	
					6	3	
							6
Fact(n-k)	5	3	2		1		
					1	1	
					2	2	
							2
Fact(n)	5		5		1		
					1	1	
					2	2	
					6	3	
					24	4	
					120	5	
							120
c := Fact(n) div (Fact(k)*Fact(n-k))			120 div (6*2)=10				

- Réécrire le programme en transformant la fonction fact à une procédure fact.

```

1 Program CombinaisonNK;
2 Uses winCRT ;
3 Var n, k, c : integer; {Variables Globales du programme}
4 Function fact (n:integer) integer ;
5 Var i : integer ; {Variables locales de la fonction fact}
6 Begin
7   F:=1;
8   For i:=1 to n do
9     f:=f*i;
10    fact:=f; {Une fonction se termine toujours par une affectation}
11  End;
12 BEGIN {Début du Programme Principal}
13  Writeln ('Donnez la valeur de n et k :');
14  Read (n, k);
15  C:= fact(n) div ( fact(k) * fact(n-k) );
16  Writeln ('La combinaison de k à partir de n = ', c);
17  End. {Fin du Programme Principal}

```

Le programme devient :

```

1 Program CombinaisonNK;
2 Uses winCRT ;
3 Var n, k, c, fn, fk, fnk : integer; {Variables Globales du programme}
4
5 Procedure fact (n:integer; var f:integer );
6 Var i : integer ; {Variables locales de la fonction fact}
7 Begin
8   F:=1;
9   For i:=1 to n do
10    f:=f*i;
11
12  End;
13 BEGIN {Début du Programme Principal}
14  Writeln ('Donnez la valeur de n et k :');
15  Read (n, k);
16
17  fact(n, fn);
18  fact(k, fk);
19  fact(n-k, fnk);
20
21  C:= fn div ( fk * fnk );
22  Writeln ('La combinaison de k à partir de n = ', c);
23  End. {Fin du Programme Principal}
24

```

Exercices supplémentaires :

```
Program exo3;
uses wincrt;
var i,n:integer;
    e:real;
FUNCTION FACT(N:integer):integer;
var i,F:integer;
begin
    F:=1;
    for i:=1 to N do
        F:=F*i;
    FACT:=F;
end;
```

```
BEGIN
    write('introduire n ');
    read(n);
    e:=1;
    for i:=1 to n do
        e:=e+1/FACT(i);
    write('Valeur de e = ',e:8:4);
END.
{ pour n = 10 on obtient e=2,7182 }
```

```
Program exo4;
uses wincrt;
TYPE TAB = array[1..100] of real;
var T:TAB;
    i,n,pos:integer;
    PG:real;

PROCEDURE PlusGrandEtPosition(T:TAB;n:integer;Var PG:real;VAR Pos:integer);
var i:integer;
begin
    PG:=T[1]; Pos:=1;
    for i:=2 to n do
        if PG<T[i] then begin
            PG:=T[i]; Pos:=i;
        end;
end;

BEGIN
    write('introduire n ');read(n);
    write('introduire les composantes de T '); for i:=1 to n do read(T[i]);
    PlusGrandEtPosition(T,n,PG,Pos);
    write('plus grand= ',PG,' son rang est ',Pos);
END.
```