

### Exercice 1

Exécuter le programme suivant :

```
Program Exo_1;
```

```
Uses winCRT ;
```

```
Var a, b, c : real; {Variables Globales du programme}
```

```
Procédure Proc1(x:real ; y:real ; var s:real) ; {sous-programme procédure n° 1}
```

```
Begin
```

```
s := x/y ;
```

```
End;
```

```
Procédure Proc2(x:real ; y:real ; s:real); {sous-programme procédure n° 2}
```

```
Begin
```

```
s := x/y;
```

```
End;
```

```
BEGIN {Début du corps Programme Principal}
```

```
a:=10; b:=5; c:=0;
```

```
Proc1(a, b, c);
```

```
Writeln('La somme est : ', c);
```

```
a:=10; b:=5; c:=0;
```

```
Proc2(a, b, c);
```

```
Writeln('La somme est : ', c);
```

```
End. {Fin du Programme Principal}
```

- Quelle est la différence entre les deux procédures Proc1 et Proc2 ?
- Quels sont les paramètres à passage par valeur et ceux à passage par variable ?
- Quels sont les paramètres formels des deux procédures ?
- Et quels sont les paramètres effectifs ?
- Exécuter le programme en donnant le type integer à la variable c. que se passe-t-il ? et pourquoi ?

### Exercice 2

Exécuter le programme suivant :

```
Program CombinaisonNK;
```

```
Uses winCRT ;
```

```
Var n, k, c : integer; {Variables Globales du programme}
```

```
Function fact(n:integer):integer ;
```

```
Var f, i : integer ; {Variables locales de la fonction fact}
```

```
Begin
```

```
f:=1;
```

```
For i:=1 to n do
```

```
f:=f*i;
```

```
fact:=f; {Une fonction se termine toujours par une affectation}
```

```
End;
```

```
BEGIN {Début du Programme Principal}
```

```
Writeln('Donnez la valeur de n et k :');
```

```
Read(n, k);
```

```
c:= fact(n) div ( fact(k) * fact(n-k) );
```

```
Writeln('La combinaison de k à partir de n = ', c);
```

```
End. {Fin du Programme Principal}
```

- Dérouler le programme ci-dessus pour  $n = 5$  et  $k = 3$
- Réécrire le programme en transformant la fonction fact à une procédure fact.

**Exercices supplémentaires :**

**Exercice 3** Ecrire un programme qui calcule la somme  $e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ , Sachant que le programme fait appel à une **fonction** FACT(n) pour calculer n!. Ecrire cette fonction et l'insérer dans le programme. Afficher le résultat dans le programme principal. (le résultat doit produire la valeur de l'exponentielle).

**Exercice 4** Ecrire un programme qui lit un **tableau** T de N réels, fait appel à une **procédure** qui détermine le plus grand élément du tableau ainsi que sa position (son rang dans le tableau). Ecrire cette procédure et l'insérer dans le programme. Afficher les résultats dans le programme principal.

```
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.
```