

TP 1 :INSTRUCTIONS REPETITIVES (LES BOUCLES)

But du TP : Manipulation de la boucle **FOR** .

Utilisation de la boucle **WHILE**.

L'idée de la boucle FOR : on répète une suite d'instructions un certain nombre de fois, donc c'est un compteur. La syntaxe est la suivante :

```
for <indice>:=<vi> to <vf> do
  BEGIN
    <instruction(s)>;
  END;
```

Attention les instructions de la boucle ne sont pas exécutées si <vi> est supérieure à <vf>.

L'idée de la boucle WHILE : on répète une suite d'instructions tant qu'une certaine condition reste vraie. La syntaxe est la suivante :

```
WHILE condition DO
  BEGIN
    instruction B ;
  END;
```

ATTENTION : Ce type de boucle peut ne jamais s'achever si la condition reste toujours vraie. Assurez vous bien que celle-ci devient donc fausse au bout d'un certain temps!!!

Exercice 1 :

1. Que fait la boucle for suivante ?

```
PROGRAM boucle ;
  VAR i:INTEGER;
  BEGIN
    FOR i:=1 TO 10 DO WRITELN(i);
  END.
```

2. Reprendre ce programme en remplaçant WRITELN(i) par WRITELN(2*i).

Exercice2:

```
PROGRAM debut;
VAR x:REAL ;
BEGIN
  x:=0;
  WHILE x <= 5 DO
  BEGIN
    x:=x+1;
    WRITELN(x);
  END;
END.
```

1. Que fait le programme ?
2. Remplacer alors la boucle WHILE par la boucle FOR.

Exercice 3 :

```
PROGRAM suite;
VAR u :REAL;
    i: INTEGER;
BEGIN
  u:=0;
  WRITELN(u);
  FOR i:= 1 TO 6 DO
  BEGIN
    u:= 2* u +3;
    WRITELN(u);
  END;
END.
```

1. Deviner l'affichage du programme précédent après exécution et reconnaître la suite mise en jeu.
2. Transformer le programme pour qu'il n'affiche que u_{100} (et rien d'autre).

Exercice 4 :

```
PROGRAM factorielle;
VAR n,k,fact:INTEGER;
BEGIN
  ..... ;
  ..... ;
  fact:= ..... ;
  K:=1;
  WHILE k<= n DO
  BEGIN
    .....;
    ..... ;
  END ;
  WRITELN(n, '!= ' , fact);
END.
```

1. Compléter le programme ci-contre pour qu'il demande un entier n , et retourne le résultat $n!$ ($n! = 1*2*3*.....*n$).
2. Compléter ensuite le programme pour mettre un garde-fou c'est-à-dire un message d'erreurs si des valeurs aberrantes (invalides) de n sont rentrées.

Exercice 5:

Multiplication par additions successives

Ecrire un algorithme qui calcule la multiplication du réel a par un entier n en n'utilisant que l'addition et en utilisant la boucle WHILE. a et n sont saisis par l'utilisateur, on suppose que n est positif et il n'y a pas d'erreur de saisie. ($a*n = a+a+a+.....+a$)
n fois

