

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

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 (* vi: valeur initiale, vf : valeur finale*)
  BEGIN
    <instruction(s)>;
  END;
```

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

Exercice 1 :

```
PROGRAM suite;
Uses crt ;
VAR u :REAL;
    i: INTEGER;
BEGIN
  u:=0;
  WRITELN(u);
  FOR i:= 1 TO 6 DO
  BEGIN
    u:= 2* u +3;
    WRITELN(u); Readkey() ;
  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 u100 (et rien d'autre).
Readkey() permet d'arrêter le défilement de l'affichage jusqu'à ce qu'on tape une touche du clavier.

Exercice 2 :

Soit la suite u définie par $U_1 = 1$ et pour tout $n \in \mathbb{N}^*$, $U_{n+1} = U_n + \frac{1}{n+1}$. Compléter et exécuter le programme pour qu'il calcule et affiche le terme U_{30}

```
Program suite ;
Uses crt ;
Var i :..... ;
    U :..... ;
Begin
  U:=.....;
  For i:=.....to..... do U:=.....;
  Writeln('le terme U30 est:',.....) ; Readkey() ;
End.
```

Exercice 3 : programme puissance

1. Compléter le programme qui demande un réel x et un entier naturel n positif ou nul, puis qui calcule x^n .

```
PROGRAM puissance;
Uses crt;
VAR x,p : ..... ; n,k : ..... ;
BEGIN
WRITE('Donner x et n'); .....;.....;
p:=.....;
IF (n>=0) THEN
begin
  FOR ..... TO ..... DO .....
  WRITELN(x:1:2,' ^',n,'=',p:1:2);
else writeln(' valeur de n doit être positive ou nul');
Readkey() ;
END.
```

2. modifier le programme précédent afin d'accepter les n négatifs (si $n < 0$ on a $x_n = 1/(x-n)$).

Exercice 4:

```
PROGRAM factorielle;
Uses crt;
VAR n,k,fact : INTEGER;
BEGIN
WRITELN('Entrer un entier n positif') ; ..... ; fact:= ..... ;
FOR k:=1 TO ..... DO ..... ;
WRITELN(n, '!= ', fact);
Readkey() ;
END.
```

1. Compléter le programme ci-contre pour qu'il demande un entier n, et retourne le résultat n!.
2. Compléter ensuite le programme pour mettre un garde-fou c'est-à-dire un message d'erreur si des valeurs aberrantes de n sont rentrées.
3. Tester le programme avec n = 10, puis n = 13, et n = 20. Que se passe-t-il ? Comment résoudre ce problème ?

Exercice 5 :

Soit la suite définie par $\{U_1 = 10; U_{n+1} = 2.U_n - 3\}$. Compléter et exécuter le programme qui demande une valeur de n à l'utilisateur et qui affiche les n premiers termes de cette suite.

```
program suite;
uses crt;
  var U, i, n : .....;
begin
writeln('Combien de nombres de la suite voulez-vous afficher ?');
readln(.....);
U:= .....;
for i:=..... to ..... do
begin
  writeln ('Le terme numéro ', i , ' de la suite est : ', .....);
  U := .....;
end;
Readkey() ;
end.
```

Exercice 6 :

Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses trois chiffres.

Exemple:153 est cubique car $153 = 1^3 + 5^3 + 3^3$.

Compléter le programme Pascal **NBR_CUBE** qui cherche et affiche tous les entiers cubiques de trois chiffres (c'est-à-dire tous les entiers cubiques compris entre 100 et 999).

```
Program Nbr_Cube;
Uses crt;
Var k, c, d, u : .....;
Begin
FOR k:=..... To ..... Do
Begin
c:=.....;
d:=.....;
u:=.....;
IF ..... = K
Then WriteLn (....., ' est un nombre cubique');
End;
End.
```