

TP n° 3**Les structures répétitives****WHILE - DO (tant que - faire)**

```
while < Condition > do  
begin  
< Instruction / bloc d'instructions >  
end;
```

REPEAT - UNTIL (répéter - jusqu'à ce que)

```
repeat  
< Instruction / Bloc d'instructions >  
until <Condition>
```

FOR - DO (pour - faire)

```
for i:=<valeur initiale> to <valeur finale> do  
begin  
< Instruction / Bloc d'instructions >  
end ;
```

```
for i:=<valeur finale > downto <valeur initiale>  
do  
begin  
< Instruction / Bloc d'instructions >  
end ;
```

Exercice 1 :

Écrire un programme qui permet de lire **N** notes, puis retourne le nombre de notes qui sont supérieures à 10. La note est comprise entre 0 et 20.

Exercice 2 :

Écrire un programme qui permet de lire un nombre entier et de retourner le nombre de chiffres qui le composent, puis afficher la somme de ses chiffres.

Exemple : N = 123456 composé de 6 chiffres, la somme est 21.

Exercice 3:

Ecrire un programme qui demande à l'utilisateur d'entrer la valeur d'une variable réelle comprise entre deux bornes, un min et un max, et qui renvoie cette valeur. Le programme doit afficher un message d'erreur si la valeur saisie n'est pas comprise entre les bornes et redemander à l'utilisateur une nouvelle saisie jusqu'à pouvoir renvoyer une valeur correcte.

Exercice 4:

Écrire un programme qui permet de lire un nombre entier et d'inverser l'ordre de ses chiffres.

Exemple : $N = 123456$ l'inverse est 654321.

Exercice 5 :

Écrire un programme qui effectue la division de deux entiers par des soustractions successives et d'afficher le reste de la division.

Exercice 6 :

Ecrire un programme qui permet de convertir un entier N écrit sous forme binaire en sa valeur décimale.