Université Batna2 Le / /2020
Faculté Technologie
Département LMD ST Informatique2 1iere année

TP 2 :INSTRUCTIONS REPETITIVES (LES BOUCLES CONDITIONNELLES)

But du TP : Manipulationtion des boucles **WHILE et REPEAT.**

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(s) >;
END;
```

Le bloc d'instruction sera répété jusqu'à ce que la condition ne soit pas vérifiée.
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!!!

L'idée de la boucle REPEAT : on répète une suite d'instructions jusqu'à ce qu'une certaine condition soit vraie. La syntaxe est la suivante : **REPEAT**

```
 < Instruction(s) >;
UNTIL <condition>;
```

- Ici, il n'est pas nécessaire d'encadrer les instructions qui se répètent par « **Begin** » et « **End ;** » ; elles sont délimitées par REPEAT et UNTIL.
- Le bloc d'instruction sera répété jusqu'à ce que la condition soit vérifiée.
- Au contraire de la boucle « **TantQue »**, l'utilisation de la boucle « **Répéter... Jusqu'à »** garantit que le bloc sera exécuté **au moins une fois** puisque le test a lieu après son exécution.

-La signification de la condition n'est plus la même :
• Avec « **TantQue »** la Condition est pour continuer la répétition (test d'entrée).
• Avec « **Répéter... Jusqu'à »** la Condition est pour arrêter la répétition (test de sortie).
-. On utilise généralement les instructions **While** ou **Repeat** lorsque l'on ne connaît pas, à l'avance, le nombre d'itérations

Exercice 1 :

```
PROGRAM debut;
  Uses crt;
 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 REPEAT UNTIL.

Exercice 2 :

On s'intéresse au jeu suivant : la machine choisit au hasard un entier entre 0 et 7 et l'utilisateur doit le deviner. Compléter le programme suivant afin de pouvoir y jouer :

```
PROGRAM jeu;
  VAR rlt, k : INTEGER;
BEGIN
   RANDOMIZE;
   rlt:=RANDOM(8); { On peut générer un nombre entier pseudo-aléatoire compris entre 0 et (8-1)=7 grâce à la
                     fonction  Random(8)  }
   REPEAT
      WRITE(' Entrez votre proposition') ;
      ..... ;
  UNTIL ..... ;
  WRITELN('vous avez gagné');
END.
```

2. Ajouter une variable qui compte le nombre d'essais nécessaire à l'utilisateur pour gagner, puis l'afficher.
3. Comment transformer le programme pour utiliser la boucle WHILE ?

Exercice 3 :

Soit la suite définie par {$U_1 = 10$; $U_{n+1} = 2.U_n - 3$}.

1-Compléter le programme qui demande une valeur de n à l'utilisateur et qui affiche les n premiers membres de cette suite.

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

2-réécrire le programme en utilisant la boucle REPEAT.

Exercice 4 :

Soit la suite définie par : S=1+1/3+1/5+1/7+1/9+……

1-Compléter le programme qui calcule la valeur de S en s'arrêtant lorsque le terme 1/x est plus petit que ε ; ε est un nombre réel <1, donné par l'utilisateur.

```
PROGRAM exo4;
  use crt ;
  var s, e: real;
BEGIN
   write ('epsilon = ');
   readln (……);
   s := ……;
  i := ……;
  repeat
    s := ……;
    i := …… ;
 until ……;
writeln ('S = ',……);
END.
```

2- Réécrire le programme en utilisant le WHILE.

Exercice 5 :

Compléter le programme qui permet de déterminer la somme des chiffres d'un entier n donné (exemple pour le nombre n=55231 il donne 1+3+2+5+5=16)

```
 Program Som_Chiffres;
 Uses    crt;
 Var     n, som, r : ……;
 Begin
    Writeln ('Donner un entier'); Readln (……); som:=……;
    Repeat
        r:= ……; som:=……; n:= ……;
    Until ……;
Writeln ('La somme de chiffres est : ', ……);
End.
```