

Chapitre 9

Les Structures de contrôle itératives

Leçon

Structures itératives complètes

Activité :

Ecrire un algorithme intitulé MESSAGE permettant d'afficher 5 fois le message "BONJOUR".

Solution :

- 0) Début MESSAGE
- 1) Ecrire("BONJOUR")
- 2) Ecrire("BONJOUR")
- 3) Ecrire("BONJOUR")
- 4) Ecrire("BONJOUR")
- 5) Ecrire("BONJOUR")
- 6) Fin MESSAGE

Constatation : D'après cette solution, on constate que l'algorithme **MESSAGE** comporte une répétition.

Solution : Pour éviter ce problème de répétition, on utilise une autre structure appelée **structure itérative** ou **répétitive** (boucle)

I- Définition :

Une définition itérative complète permet de répéter l'exécution d'une suite d'instructions un nombre de fois connu l'avance.

II-Vocabulaire et syntaxe :

Au niveau de l'analyse et de l'algorithme	Au niveau pascal
<p>[Init]</p> <p>Pour C de Vi à Vf Répéter</p> <p style="padding-left: 40px;">instruction 1</p> <p style="padding-left: 80px;">instruction 2</p> <p style="padding-left: 40px;">.....</p> <p style="padding-left: 40px;">instruction n</p> <p>FinPour</p>	<p>init ;</p> <p>For C:=Vi TO Vf DO</p> <p>Begin</p> <p style="padding-left: 40px;">instruction 1;</p> <p style="padding-left: 40px;">instruction 2;</p> <p style="padding-left: 40px;">.....</p> <p style="padding-left: 40px;">instruction n;</p> <p>End ;</p>

- ✓ **C** : variable de type scalaire (entier ou caractère) permet de compter le nombre de répétitions effectuées. L'incréméntation (+) de 1 de la valeur de C se fait automatiquement.
- ✓ **Vi** : valeur initiale du compteur
- ✓ **Vf** : valeur finale que doit atteindre le compteur, c'est la dernière valeur pour laquelle le traitement effectué

Activité :

Ecrire une analyse, un algorithme puis un programme en Pascal intitulé **SAISIETAB**, permettant le remplissage d'un tableau **T** avec **N** réels puis d'afficher les nombres paires de ce tableau.

Analyse

```

Resultat = affichage
Traitemnet=
Affichage = []
  Pour i de 1 à n faire
    Si T[i] mod 2 = 0 alors
      Ecrire (T[i])
    Fin si
  Fin pour
T=[]
  Pour i de 1 à n faire
    T[i]=donnée("donner l'element n° ",i)
  Fin pour
i=compteur
n=donnée("Donner la taille du tableau")
Donnée : n
    
```

Tableau de déclaration des objets

Objet	Nature / Type	Rôle
i	Entier	Compteur
n	Entier	Taille de T
T	Tableau de n entiers	Contenir des entiers

Algorithme

- 0) Debut SAISIETAB
- 1) ecrire("Donner la taille du tableau"), lire(n)
- 2) Pour i de 1 à n faire
 ecrire("donner l'element n° ",i), lire(T[i])
 Fin pour
- 3) Pour i de 1 à n faire
 Si $T[i] \bmod 2 = 0$ alors
 Ecrire (T[i])
 Fin si
 Fin pour
- 4) Fin SAISIETAB

Traduction pascal

```
program SAISIETAB;  
uses winCRT;  
var  
T:array[1..100] of integer;  
n,i:integer;  
begin  
writeln('Donner la taille du tableau');  
readln(n);  
for i:=1 to n do  
begin  
writeln('Donner l'élément n° ',i);  
readln(T[i]);  
end;  
for i:=1 to n do  
if T[i] mod 2 = 0 then  
writeln(T[i]);  
end.
```

II-Parcours décroissant :

Parfois, on se trouve contraint de faire une itération avec un comptage décroissant.

On exprime cette structure comme suit :

R = [] Pour c de n à 1 (pas = -1) Faire

 Instruction 1
 Instruction 2
 ...
 ...
 Instruction p
FinPour

NB

Comme on peut incrémenter c de 1, on peut le décrémenter de 1 sauf qu'il faut modifier la structure en Turbo Pascal par :

FOR C: = Vf DOWNTO Vi DO

BEGIN

Instruction 1;

.....

Instruction n;

END;

III- Problèmes récurrents :

Souvent, certains résultats sont définis d'une façon récurrente. Dans de tels cas, le résultat se forme au fur et à mesure et à une étape donnée, il dépend d'un certain nombre de résultats précédents. Une telle structure est dite itérative récurrente. En fait, les résultats intermédiaires sont gérés par une relation de récurrence. Si cette relation lie deux éléments successifs, on dit que la récurrence est d'ordre 1, si elle lie trois éléments successifs, on dit qu'elle d'ordre 2, etc.

Activité :

Ecrire un programme qui saisit un tableau T2 unidimensionnel de N éléments , représentant les moyennes des élèves d'une classe, puis calculer et afficher la moyenne de la classe

Analyse

Resultat= écrire("La moyenne de la classe est ",MoyC)

Traitement=

MoyC \leftarrow SommeMoy/n

SommeMoy=[SommeMoy \leftarrow 0]

Pour i de 1 à n faire

SommeMoy \leftarrow SommeMoy + T2[i]

Fin pour

T2=[]

Pour i de 1 à n faire

T2=donnée("donner l'element n° ",i)

Fin pour

i=compteur

n=donnée("Donner la taille du tableau")

Donnée : n

Tableau de déclaration des objets

Objet	Nature / Type	Rôle
i	Entier	Compteur
n	Entier	Taille de T
T	Tableau de n réelles	Contenir les moyennes des élèves
SommeMoy	Réelles	Somme de moyennes
MoyC	réelles	Moyenne de la classe

Algorithme

0) Debut Moyenne_classe

1) écrire("Donner la taille du tableau"), lire(n)

2) Pour i de 1 à n faire

écrire("donner l'element n° ",i), lire(T2[i])

Fin pour

3) SommeMoy \leftarrow 0

Pour i de 1 à n faire

SommeMoy \leftarrow SommeMoy + T2[i]

Fin pour

4) MoyC \leftarrow SommeMoy/n

5) écrire("La moyenne de la classe est ",MoyC)

6) Fin Moyenne_classe

Traduction Pascal

```
program Moyenne_classe;
uses wincrt;
var
T2:array[1..100] of real;
n,i:integer;
SommeMoy,MoyC:real;
begin
writeln('Donner la taille du tableau');
readln(n);
for i:=1 to n do
begin
writeln('Donner l"élément n° ',i);
readln(T2[i]);
end;
SommeMoy:=0;
for i:=1 to n do
SommeMoy:=SommeMoy+T2[i];
MoyC:=SommeMoy/n;
writeln('la moyenne de la classe est ',MoyC);
end.
```