

Chapitre 10 : Les structures de contrôle conditionnelles

Introduction

On a vu dans le chapitre précédent les différentes structures simples. Qui sont les opérations d'entrée sortie et l'affectation.

Dans ce chapitre nous allons étudier les structures de contrôle conditionnelles simples, généralisées et à choix.

I- Les structures conditionnelles simples :

1) Activité 1

1/ Lire les phrases suivantes.

Si je révise bien mes cours **alors** j'aurais une bonne note.

Si la pluie va tomber **alors** on restera à la maison, **sinon** on ira à la pêche.

Si j'ai beaucoup d'argent **alors** j'achèterais une voiture neuve **sinon** j'achèterais une voiture occasion.

⇒ Il y a une condition à respecter qui débute par SI, une première action à faire si la condition est vérifiée débuté par ALORS et une deuxième action à faire si la condition n'est pas vérifiée précédée par SINON.

2) Syntaxes

Algorithmique	Pascal
SI condition ALORS Traitement 1	If condition then Begin Traitement 1 ;
SINON Traitement 2	End
FINSI	Else Begin Traitement 2 End ;

Remarques :

- ▶ Dans certains cas, on trouve la structure conditionnelle ainsi : **SI** condition **ALORS** Traitement
- ▶ La condition doit avoir une valeur booléenne, et elle peut être une condition composée.
- ▶ Pour la traduction en Pascal, si le traitement contient une seule instruction, on peut éliminer le Begin et le End.
- ▶ Dans la structure conditionnelle, on exécute soit le traitement 1 soit le traitement 2, mais pas les deux à la fois.

3) Activité 2

Ecrire l'analyse et l'algorithme d'un programme qui permet d'afficher le résultat final d'un élève en fin d'année. La décision est "admis" si sa moyenne ≥ 10 sinon "refusé". La moyenne est donnée par l'utilisateur



-- FIN

4) Application 1

Ecrire un programme qui permet de saisir un réel X et d'afficher « Ce nombre est positif » s'il est positif négatif sinon.

5) Application 2

Ecrire un programme qui permet de saisir un entier N et d'afficher s'il est pair ou impair.

II- Les structures conditionnelles généralisées :

1) Activité 3



On se propose d'écrire l'algorithme d'un programme qui permet d'afficher la mention d'un élève. Rappelons que pour déterminer la mention, on suit la démarche suivante :
 Si $10 \leq \text{Moy} < 12$ alors la mention est « **Passable** » sinon si $12 \leq \text{Moy} < 14$ alors la mention est « **Assez bien** » sinon si $14 \leq \text{Moy} < 16$ alors la mention est « **bien** » sinon la mention est « **très bien** »

2) Syntaxe

Algorithmique

```

SI condition1 ALORS
  Traitement 1
SINON SI condition 2 ALORS
  Traitement 2
SINON SI condition 3 ALORS
  Traitement 3
.
.
.
SINON SI condition n ALORS
  Traitement n
SINON traitement n+1
FINSI
  
```

-- FIN

Pascal

```

If condition1 then
  Begin
    Traitement 1 ;
  End
Else if condition2 then
  Begin
    Traitement 2
  End
Else if condition3 then
  Begin
    Traitement 3
  End ;
...
Else if condition n then
  Begin
    Traitement n
  End
Else
  Begin
    Traitement n+1
  End ;
  
```

3) Application

Faire l'analyse et l'algorithme d'un programme qui, à partir du numéro d'un mois permet d'afficher la saison.

Les mois sont numérotés de 1 à 12 en commençant par Janvier.

III- Les structures conditionnelles à choix :

Supposons qu'on a un distributeur de boisson (café, jus, thé, eau minérale, boisson gazeuse). Pour avoir de la boisson, il faut appuyer sur la touche convenable à cette boisson.

Comment ce distributeur va fonctionner ?

1) Activité 5

Ecrire l'analyse d'un programme qui permet de saisir deux réels a et b et un opérateur et d'afficher le résultat de l'opération choisie.

Nom : operation

Résultat = écrire (R)

R = Si (op == '+') alors

R ← a + b

Sinon Si (op == '-') alors

R ← a - b

Sinon Si (op == '*') alors

R ← a * b

Sinon R ← a / b

finSi

a = Donnée (" Donner le premier réel ")

b = Donnée (" Donner le deuxième réel ")

op = Donnée (" Donner l'opérateur ")

⇒ Si on saisit l'opérateur '/' par exemple, l'ordinateur sera obligé de tester le +, le - et le *.

2) Syntaxe

Syntaxes	
En analyse et en algorithme	En Pascal
Selon Sélecteur Faire Valeur(s)1 : Traitement 1 Valeur(s)n : Traitement n Sinon Traitement m Fin Selon	Case Sélecteur OF Valeur(s)1 : Traitement 1 ; Valeur(s)n : Traitement n Else Traitement m ; End ;

Rques :

- Le sélecteur doit être de type **scalaire (entier, caractère)**.
- Valeur(s)₁, valeur(s)₂ ... Valeur(s)_n peuvent se présenter sous forme d'une seule valeur, de listes dont les valeurs seront séparées par la virgule ou même sous forme d'intervalles.

Exp :

1, 3, 5, 7 : traitement ou 1..10 : traitement

- La partie « **Sinon** » est facultative.
- Chaque traitement peut être composé d'une ou de plusieurs instructions. Dans ce dernier cas, il est nécessaire d'indiquer les instructions entre « **Begin** » et « **End ;** » en Turbo Pascal.

-- **FIN**

3) Application

Ecrire l'analyse et l'algorithme d'un programme qui, à partir du numéro d'un mois permet d'afficher le nombre de jours qui lui correspond.

IV- Applications

Exercice 1

Ecrire une analyse , un algorithme et un programme pascal qui lit puis résout une équation de second degré à coefficients réels.

-- **FIN**

Exercice 2

On veut saisir un caractère c et une chaîne de caractère ch., puis afficher si le caractère c appartient à la chaîne ch. ou non .

Exercice 3

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

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

Ecrire un programme qui permet de vérifier si un nombre est cubique ou non

Faire l'analyse , algorithme et Pascal.

Exercice 4

Ecrire un programme qui permet de chercher puis d'afficher le plus grand de deux entiers donnés

Ensuite le plus grand de 3 entiers donnés.

Exercice 5

Ecrire une analyse , un algorithme et un programme Pascal qui saisit deux dates sous la forme suivante « jj/mm/aaaa » puis affiche la date la plus récente.