

LES STRUCTURES DE CONTROLE CONDITIONNELLES

I. INTRODUCTION

Activité1 :

Écrire une analyse, un algorithme et sa traduction en Turbo Pascal du programme nommé **PARITE**, qui permet de lire un nombre NB et affiche le message "Pair" si NB est pair sinon "Impair"

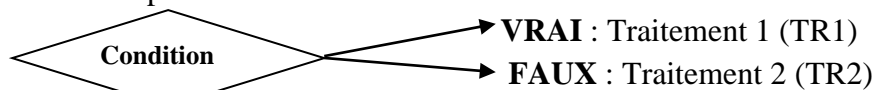
Préanalyse : NB → Traitement → Message Avec Message = ou

Il faut choisir entre deux possibilités :

- Message = "Pair" si
- Message = "Impair" si.....

Remarque : le choix dépend du résultat d'une expression de comparaison (résultat ou). On dit que l'action à exécuter dépend d'une **CONDITION**.

Dans telle situation, on doit faire recours à une structure permettant à un programme de choisir entre possibilités de traitement suivant le résultat d'évaluation d'une condition.



Cette structure est dite : **structure de contrôle CONDITIONNELLE**

Il existe deux formes d'instructions conditionnelles :

- Forme simple
- Forme généralisée

II. FORME SIMPLE

II.1. Forme complète

• **Définition:** Elle consiste à choisir le traitement à exécuter selon la valeur d'une

Exemple : **Si** je réussis bien mon baccalauréat **alors** j'aurai mon premier choix, **sinon** j'espère avoir mon deuxième choix.

• **Vocabulaire et syntaxe:**

<i>Au niveau de L'analyse</i>	<i>Au niveau de L'algorithme</i>	<i>Au niveau du Turbo Pascal</i>
[Init] Si condition Alors Instruction 1 de TR1 Instruction n de TR1 Sinon Instruction1 de TR2 Instruction n de TR2 FinSi {Init} Si condition Alors Instruction 1 de TR1 Instruction n de TR1 Sinon Instruction1 de TR2 Instruction n de TR2 Finsi ;{Init} If condition (s) THEN BEGIN Instruction 1 de TR1 ; ; Instruction n de TR1 ; END ELSE BEGIN Instruction1 de TR2 ; ; Instruction n de TR2 ; END ;

• **Remarque**

- [init] est une séquence d'instruction qui contiendra les éventuelles
- Dans le cas de plusieurs instructions pour TR1 et TR2, on doit obligatoirement utiliser en Pascal les délimiteurs (**BEGIN** et **END;**)
- **Attention** : Il n'y a pas de ";" après l'instruction qui précède **ELSE** (**END**) car l'instruction **IF** n'est pas encore finie.
- **Fin si** en Pascal.
- Condition(s) peut être simple ou complexe (combine plusieurs conditions simples, chacune entre parenthèses, à l'aide des opérateurs logiques).

Exemple : Si (a>b) et (c<d) alors

• **Application 1**

- Quelles erreurs ont été commises dans chacune des instructions suivantes:
 - a) if a<b then x := x+1 ; else x := x-1;
 - b) if a<b then x :=x+1; y :=b end else x :=x-1; y :=a end
 - c) if n := 0 then p = 1 ;

Application 2

Ecrire l'analyse et l'algorithme d'un programme permettant de remplacer le 1^{er} caractère d'une chaîne donnée par son code ascii s'il est lettre miniscule et par son rang dans l'alphabet s'il est lettre majuscule.

Conclusion : La partie **Sinon** dans une structure conditionnelle simple n'est pas toujours Dans le cas où elle est absente, on dit que la structure de contrôle conditionnelle a une forme simple réduite.

II.2. Forme simple réduite

- **Définition:** Elle se réduit à l'exécution d'un **traitement** quand une **condition** est réalisée.

Exemple : Si je fais des études de médecine **alors** je servirai l'humanité.

• **Vocabulaire et syntaxe:**

<i>Au niveau de L'analyse</i>	<i>Au niveau de L'algorithme</i>	<i>Au niveau du Turbo Pascal</i>
[Init] Si condition (s) Alors	{Init} Si condition (s) Alors	{Init} IF condition (s) THEN
Instruction 1 de TR1	Instruction 1 de TR1	BEGIN
.....	Instruction 1 de TR1 ;
Instruction n de TR1	Instruction n de TR1 ;
Finsi	Finsi	Instruction n de TR1 ;
		END;

• **Remarque**

Lorsque l'évaluation de la condition produit la valeur :

- **VRAI** : les instructions entre **Alors** et **Finsi** sont exécutées
- **FAUX** : les instructions entre **Alors** et **Finsi** ne sont pas exécutées

Application 3

- Présenter l'analyse et l'algorithme permettant:
 - De vérifier si une chaîne contient des chiffres ou non.
 - De saisir un mot et d'afficher si un caractère c existe dans ce mot ou non.
 - De saisir deux entiers au clavier puis afficher leur maximum.
 - De saisir la moyenne d'un élève et d'afficher est ce que l'élève a réussi ou non

III. FORME GENERALISEE

Activité 2 :

Écrire une analyse, un algorithme et sa traduction en Turbo Pascal du programme nommé **Mention**, qui permet d'afficher la mention d'un élève (redouble, passable, bien ou très bien).

REMARQUES : Dans ce cas l'utilisation d'une seule structure conditionnelle à forme complète (Permet de choisir entre deux possibilités de traitement) : Donc nous ferons appel à la forme généralisée imbriquée : les instructions qui suivent **Alors** ou bien **Sinon** peuvent être elles même des structures conditionnelles.

• **Définition :** C'est une structure algorithmique qui fait appel àtraitements. L'exécution d'un traitement entraîne automatiquement laexécution des autres traitements

• Vocabulaire et syntaxe

Au niveau de l'analyse	Au niveau Pascal
[Init]	{Init}
Si <i>condition1</i> alors	If <i>condition1</i> then
Traitement1	Begin Traitement1 End
Sinon si alors	Else if
.....	Else if <i>conditionN-1</i> then
Sinon si <i>conditionN-1</i> alors	Begin TraitementN-1 end
TraitementN-1	Else
Sinon	Begin TraitementN End ;
TraitementN	
FinSi	

• Remarque

Lorsque l'évaluation de condition1 (s) (C1) produit la valeur :

- **VRAI** : seul le traitement1 (TR1) est exécuté
- **FAUX** : on passe à l'évaluation de condition2 (s) (C2), si elle produit la valeur :
 - VRAI** : seul le traitement2 (TR2) est exécuté
 - FAUX** : on passe à l'évaluation de condition3 (s) (C3) et ainsi de suite...

Si aucune de N – 1 (pour notre exemple N = 4) premières conditions ne produit la valeur VRAI, par conséquent le traitement N (TR4) est exécuté.

• Application 4 :

Écrire une analyse, un algorithme et sa traduction en Turbo Pascal du programme nommé **COMMISSION**, qui permet de lire le montant total des ventes **V** puis affiche la commission **C** correspondante. La commission est calculée de la manière suivante :

- $V \leq 50 \rightarrow C = 0$ ①
- $50 < V \leq 500 \rightarrow C = 10 \% V$ ②
- $500 < V \leq 1000 \rightarrow C = 50 + 15\% V$ au dessus de 500 ③
- $V > 1000 \rightarrow C = 200 + 20\% V$ au dessus de 1000 ④

• Application 5 :

Présenter l'analyse et l'algorithme permettant d'afficher le nombre de jour d'un mois donné.

IV. LA STRUCTURE CONDITIONNELLE A CHOIX

REMARQUE

L'analyse montre un choix parmi plusieurs possibilités suivant le contenu que prendra le sélecteur C (variable de type caractère : un) en utilisant la structure conditionnelle imbriquée (plusieurs instructions Si ...) : Dans telle situation il est plus intéressant de faire recours à la structure de choix qui évite l'utilisation d'une trop longue structure conditionnelle généralisée et permet une meilleur de la solution.

• **Vocabulaire et syntaxe:**

Au niveau de L'analyse et l'algorithm	Au niveau du Turbo Pascal
<p>[Init] Selon Sélecteur Faire</p> <p>Valeur1 : Action 1</p> <p>Valeur2 : Action 2-1</p> <p>.....</p> <p> Action 2-n</p> <p>Valeur4, Valeur5, Valeur6 : Action 4</p> <p>Valeur7 .. Valeur20 : Action5</p> <p>Sinon</p> <p> Action n+1</p> <p>Finsel</p>	<p>[Init] Case Sélecteur Of</p> <p>Valeur1 : Action 1 ;</p> <p>Valeur2 : BEGIN</p> <p> Action 2-1</p> <p> </p> <p> Action 2-n</p> <p> END;</p> <p>Valeur4, Valeur5, Valeur6 : Action 4 ;</p> <p>Valeur7 .. Valeur20 : Action5 ;</p> <p>Else</p> <p> Action n+1 ;</p> <p>End ;</p>

REMARQUES

- Dans tous les cas Sélecteur doit être de type (elle ne peut jamais être de type réel ou chaîne de caractère)
- Sélecteur est comparée à :
 - Une seule valeur
 - Une série de valeurs séparée par ","
 - Un intervalle
- En cas d'égalité du sélecteur à une valeur (ou fait partie d'un intervalle) l'action qui lui fait associer est exécutée. Les autres ne seront pas exécutées
- En cas d'inégalités il y aura exécution de la partie Sinon (Action n+1) si elle est présente (optionnelle)
- Si la valeur du sélecteur appartient à plusieurs valeurs (valeur i, valeur i+1, ...), seule la première valeur appartenant à valeur i sera exécutée.

Application 6

Ecrire un algorithme **SELECTION** qui lit un caractère C et affiche :

- Son majuscule si C en minuscule
- Son minuscule si C en majuscule
- La racine carrée si C représente un chiffre
- Le code ASCII de C s'il est un caractère de ponctuation :., ?!;
- "Caractère non valide" dans le cas contraire

Application 7 :

Ecrire un algorithme du programme « **DATE** », qui saisit une date sous la forme jj/mm/aaaa (chaîne de 10 caractère) où jj représente le jour, mm le mois, aaaa l'année et affiche un commentaire sur la personne ayant cette date de naissance (jeune, adulte,...).

