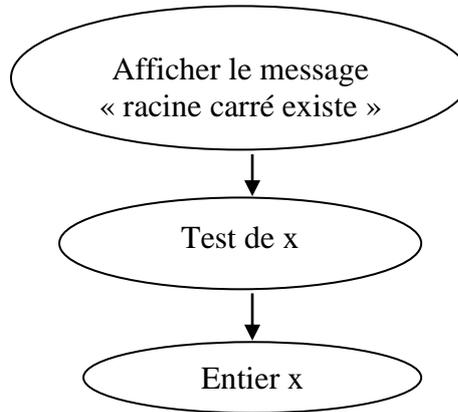


**1** Forme simple réduite

**a /** **Activité**

Ecrire un programme pascal qui permet de saisir un entier x et d'afficher le message « racine carré existe » si x est positif 0.

**Pré-analyse**



**Analyse**

**Grille d'analyse**

NOM : RACINE_CARRE		
S	L.D.E	O.U
3	<b>Résultat</b> = Écrire (message)	message x
2	message = [message ← "" ] Si x > 0 alors message ← "racine carré existe"	
1	Finsi	
4	x = Donnée ("Taper un entier : ") <b>Fin</b> RACINE_CARRE	

**Tableau de déclaration des objets**

Objet	Nature / Type
message	Chaîne de caractères
x	Entier.

## Algorithme

- 0) Début RACINE\_CARRE
- 1) Ecrire (" Taper un entier : "), lire (x)
- 2) message ← "" Si x > 0 alors  
    message ← "racine carré existe"  
    Finsi
- 3) Écrire (message)
- 4) Fin RACINE\_CARRE

## Traduction en Pascal

```
PROGRAM RACINE_CARRE;  
USES WINCRT;  
VAR  
    x : INTEGER;  
    Message : STRING;  
BEGIN  
    WRITE ("Taper un entier : ");  
    READLN (x);  
    message:= " "  
    IF x >0 THEN  
        Message := 'racine carré existe' ;  
    WRITELN (message);  
END.
```

### **b / Définition**

Une structure de contrôle conditionnelle a **une forme simple réduite** si son traitement est exécutée quand la condition est vraie.

### **c / Vocabulaire et syntaxe**

Au niveau de l'analyse et de l'algorithme	Au niveau de Turbo Pascal
..... [Init] <b>Si</b> condition <b>Alors</b> Instruction 1 Instruction 2 ..... Instruction N <b>FinSi</b> ..... .....	..... ; {Init}; <b>IF</b> condition <b>THEN</b> <b>Begin</b> Instruction 1 ; Instruction 2 ; ..... ; Instruction N ; <b>End ;</b> .....

#### **Remarque :**

- [Init] est une suite d'instructions qui contient les initialisations.
- Condition est une expression booléenne qui peut donner soit la valeur vrai (true) soit la valeur faux (false).

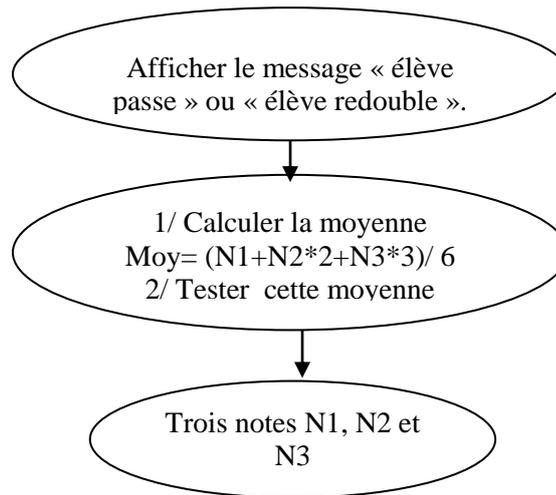
## **2 Forme simple alternative ou complète**

### **a / Activité**

Ecrire un programme Pascal qui permet de :

1. Calculer la moyenne d'un élève à partir de trois notes N1, N2 et N3 de coefficients respectifs 1, 2, et 3
2. Afficher les messages :
  - « élève passe » si sa moyenne  $\geq 10$ .
  - « élève redouble » sinon.

## Pré-analyse



## Analyse

## Grille d'analyse

NOM : DECISION		
S	L.D.E	O.U
6	<b>Résultat</b> = Écrire (message)	message
5	message = [ message ← "" ] Si moy >= 10 alors message ← "élève passe" Sinon message ← "élève redouble" FinSi	moy N1 N2 N3
4	moy ← (N1+N2*2+N3*3)/ 6	
1	N1= Donnée (" Taper la 1 <sup>ère</sup> note")	
2	N2= Donnée (" Taper la 2 <sup>ème</sup> note")	
3	N3= Donnée (" Taper la 3 <sup>ème</sup> note")	
7	Fin DECISION	

## Tableau de déclaration des objets

Objet	Nature / Type
Message	Chaîne de caractères
Moy	Réel
N1	Réel
N2	Réel
N3	Réel

## Algorithme

- 0) Début DECISION
- 1) Ecrire ("Taper la 1<sup>ère</sup> note"), lire (N1)
- 2) Ecrire ("Taper la 2<sup>ème</sup> note"), lire (N2)
- 3) Ecrire ("Taper la 3<sup>ème</sup> note"), lire (N3)
- 4) moy ← (N1+N2\*2+N3\*3) / 6
- 5) message ← [ "" ] Si moy >= 10 alors  
message ← "élève passe"  
Sinon  
message ← "élève redouble"  
FinSi
- 6) Ecrire (message)
- 7) Fin DECISION

## Traduction en Pascal

**PROGRAM** DECISION;

**USES** WINCRT;

**VAR**

N1, N2, N3, moy : REAL ;

message : STRING;

**BEGIN**

WRITELN ('Taper la première note ');

READ (N1) ;

WRITELN ('Taper la deuxième note ');

READ (N2) ;

WRITELN ('Taper la troisième note ');

READ (N3) ;

Moy:= (N1+N2\*2+N3\*3) / 6

message:= '';

**IF moy > = 10 THEN**

message := 'élève passe'

**ELSE**

message := 'élève redouble' ;

**WRITE** (message);

**END.**

### **b / Définition**

Une structure de contrôle conditionnelle a une forme complète si suivant la valeur d'une condition, on exécute soit un traitement T1 soit un traitement T2.

Si la condition est vraie, alors traitement T1 est exécuté. Dans le cas contraire, traitement T2 est exécuté.

### **c / Vocabulaire et syntaxe**

Au niveau de l'analyse et de l'algorithme	Au niveau de Turbo Pascal
..... [Init] <b>Si</b> condition <b>alors</b> Traitement 1 <b>Sinon</b> Traitement 2 <b>FinSi</b> .....	..... ; {Init}; <b>IF</b> condition <b>THEN</b> <b>Begin</b> Traitement 1 ; <b>End</b> <b>ELSE</b> <b>Begin</b> Traitement 2 ; <b>End ;</b> ..... ;

### Remarques :

1. Dans un programme Pascal, le point virgule (;) avant le ELSE est interdit.
2. Au niveau de Turbo Pascal, lorsque Traitement 1 ou traitement 2 comporte plus qu'une instruction, il faut encadrer ce traitement entre Begin et End.

## II. FORME GENERALISEE (LES TRAITEMENTS CONDITIONNELS IMBRIQUES)

### 1 Définition

Les traitements de alors et de sinon peuvent être donnés sous la forme d'un traitement conditionnel : on parle dans ce cas de traitements conditionnels imbriqués.

### 2 Vocabulaire et syntaxe

Au niveau de l'analyse et de l'algorithme	Au niveau de Turbo Pascal
<p>.....</p> <p>[Init] <b>Si</b> condition 1 <b>alors</b></p> <p style="padding-left: 40px;">Traitement 1</p> <p style="padding-left: 20px;"><b>Sinon</b></p> <p style="padding-left: 40px;"><b>Si</b> condition 2 <b>alors</b></p> <p style="padding-left: 80px;">Traitement 2</p> <p style="padding-left: 20px;"><b>Sinon</b></p> <p style="padding-left: 40px;">.....</p> <p style="padding-left: 40px;"><b>Si</b> condition n-1 <b>alors</b></p> <p style="padding-left: 80px;">Traitement n-1</p> <p style="padding-left: 20px;"><b>Sinon</b></p> <p style="padding-left: 40px;">Traitement n</p> <p><b>FinSi</b></p> <p>.....</p>	<p>..... ;</p> <p>{Init};</p> <p><b>IF</b> condition 1 <b>THEN</b></p> <p style="padding-left: 40px;">Traitement 1 ;</p> <p><b>ELSE</b></p> <p style="padding-left: 40px;"><b>IF</b> condition 2 <b>THEN</b></p> <p style="padding-left: 80px;">Traitement 2 ;</p> <p style="padding-left: 40px;"><b>ELSE</b></p> <p style="padding-left: 40px;">.....</p> <p style="padding-left: 40px;"><b>IF</b> condition n-1 <b>THEN</b></p> <p style="padding-left: 80px;">Traitement n-1</p> <p style="padding-left: 40px;"><b>ELSE</b></p> <p style="padding-left: 40px;">Traitement n ;</p> <p>..... ;</p>

### 3 Activité

Ecrire un programme pascal qui permet de :

1. Calculer la moyenne d'un élève à partir de trois notes N1, N2 et N3 de coefficients respectifs 1, 2 et 3.
2. Afficher la mention correspondante.

<i>Moyenne</i>	<i>Mention</i>
Moyenne $\geq 16$	Très bien
$14 \leq$ moyenne $< 16$	Bien
$12 \leq$ moyenne $< 14$	Assez bien
$10 \leq$ moyenne $< 12$	Passable
moyenne $< 10$	Redouble



## **Algorithme**

- 0) Début MENTION
- 1) Ecrire ("Taper la 1<sup>ère</sup> note :"), lire (N1)
- 2) Ecrire ("Taper la 2<sup>ème</sup> note :"), lire (N2)
- 3) Ecrire ("Taper la 3<sup>ème</sup> note :"), lire (N3)
- 4) Moy  $\leftarrow (N1+N2*2+N3*3) / 6$
- 5) [ ment  $\leftarrow$  "" ] Si moy  $\geq 16$  alors  
    ment  $\leftarrow$  "très bien"  
    Sinon  
        Si moy  $\geq 14$  alors  
            ment  $\leftarrow$  "bien"  
            Sinon  
                Si moy  $\geq 12$  alors  
                    ment  $\leftarrow$  "assez bien"  
                    Sinon  
                        Si moy  $\geq 10$  alors  
                            ment  $\leftarrow$  "passable"  
                            Sinon  
                                ment  $\leftarrow$  "redouble"  
                        FinSi  
            FinSi  
        FinSi
- 6) Ecrire ("La mention est : ", ment)
- 7) Fin MENTION

## **Traduction en Pascal**

```
PROGRAM MENTION;  
USES WINCRT;  
VAR  
    N1, N2, N3, moy : REAL;  
    ment : STRING;  
BEGIN  
    WRITE ('Taper la 1ère note : ');  
    READ (N1) ;  
    WRITE ('Taper la 2ème note : ');  
    READ (N2) ;  
    WRITE ('Taper la 3ème note : ');  
    READ (N3) ;  
    Moy := (N1+N2*2+N3*3) / 6;  
    ment:= '';  
    IF moy >= 16 THEN  
        ment := 'très bien'  
    ELSE  
        IF moy >= 14 THEN  
            ment := 'bien'  
        ELSE  
            IF moy >= 12 THEN  
                ment := 'assez bien'  
            ELSE  
                IF moy >= 10 THEN  
                    ment := 'passable'  
                ELSE  
                    ment := 'redouble' ;  
    END ;  
    WRITE ('La mention est : ', ment);  
END.
```