



## Fiche TD N° 02.3 : Structures Conditionnelles Solution

**Remarque :** pour chaque exercice :

- Dessiner l'algorithme (organigramme de programmation) correspondant.
- Faire en sorte que des libellés apparaissent clairement.

1. Écrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est strictement positif ou non, et affiche ce résultat.

Les conditions possibles :

1.  $N > 0$
2.  $N \leq 0$

### **Algorithme Exo1**

Var

N : réel ;

#### **Début**

Ecrire ("Ce programme vous informe si un nombre est positif ou non") ;

Ecrire ("Entrez la valeur de ce nombre") ;

Lire (N) ;

**Si** ( $N > 0$ ) **Alors**

Ecrire ("Ce nombre est strictement positif") ;

**Sinon**

Ecrire ("Ce nombre n'est pas positif") ;

**Finsi** ;

**Fin.**

OU

### **Algorithme Exo1**

```

Var
N : réel ;
Début
Ecrire ("Ce programme vous informe si un nombre est positif ou non") ;
Ecrire ("Entrez la valeur de ce nombre") ;
Lire (N) ;
    Si (N <= 0) Alors
        Ecrire ("Ce nombre n'est pas positif") ;
    Sinon
        Ecrire ("Ce nombre est strictement positif") ;
    Finsi ;
Fin.

```

2. Écrire un algorithme qui demande deux nombres m et n à l'utilisateur et l'informe ensuite si le produit est négatif ou positif. On inclut dans l'algorithme le cas où le produit peut être nul.

Les conditions possibles :

n	m	produit n*m
nul	nul	nul
	positive	nul
	négative	nul
positive	nul	nul
	positive	positive
	négative	négative
négative	nul	nul
	positive	négative
	négative	positive

### **Algorithme Exo2**

```

Var
N, M : entier ;
Début
Ecrire ("Ce programme ....") ;
Ecrire ("Entrez la valeur du premier nombre") ;
Lire (N) ;
Ecrire ("Entrez la valeur du deuxième nombre") ;

```

```

Lire (M) ;
Si (N = 0) Alors
    Ecrire ("Le produit de ces deux nombres est nul") ;
Sinon
    Si (N > 0) Alors
        Si (M = 0) Alors
            Ecrire ("Le produit de ces deux nombres est nul") ;
        Sinon
            Si (M > 0) Alors
                Ecrire ("Le produit de ces deux nombres est positive") ;
            Sinon
                Ecrire ("Le produit de ces deux nombres est négative") ;
            Finsi ;
        Finsi ;
    Sinon
        Si (M = 0) Alors
            Ecrire ("Le produit de ces deux nombres est nul") ;
        Sinon
            Si (M > 0) Alors
                Ecrire ("Le produit de ces deux nombres est négative") ;
            Sinon
                Ecrire ("Le produit de ces deux nombres est positive") ;
            Finsi ;
        Finsi ;
    Finsi ;
Finsi ;

```

**Fin.**

Les conditions possibles :

$(n = 0) \text{ ou } (m = 0) \Rightarrow \text{produit : nul}$

$((n > 0) \text{ et } (m > 0)) \text{ ou } ((n < 0) \text{ et } (m < 0)) \Rightarrow \text{produit : positive}$

$((n < 0) \text{ et } (m > 0)) \text{ ou } ((n > 0) \text{ et } (m < 0)) \Rightarrow \text{produit : négative}$

**Algorithme** Exo2

Var

N, M : entier ;

**Début**

Ecrire ("Ce programme ....") ;

Ecrire ("Entrez la valeur du premier nombre") ;

Lire (N) ;

Ecrire ("Entrez la valeur du deuxième nombre") ;

Lire (M) ;

**Si** ( $n = 0$ ) ou ( $m = 0$ ) **Alors**

Ecrire ("Le produit de ces deux nombres est nul") ;

**Sinon**

**Si** ( $(n > 0)$  et ( $m > 0$ )) ou ( $(n < 0)$  et ( $m < 0$ )) **Alors**

Ecrire ("Le produit de ces deux nombres est positive") ;

**Sinon**

Ecrire ("Le produit de ces deux nombres est négative") ;

**Finsi** ;

**Finsi** ;

**Fin.**

3. Écrire un algorithme qui demande un entier à l'utilisateur, teste si ce nombre est strictement positif, nul ou strictement négatif, et affiche ce résultat.

Les conditions possibles :

**Algorithme Exo3**

Var

N : réel ;

**Début**

Ecrire ("Ce programme vous informe si un nombre est positif, négatif ou nul") ;

Ecrire ("Entrez la valeur de ce nombre") ;

Lire (N) ;

**Si** ( $N < 0$ ) **Alors**

Ecrire ("Ce nombre n'est pas strictement négatif") ;

**Sinon**

```

Si ( $N > 0$ ) Alors
    Ecrire ("Ce nombre n'est pas strictement positif");
Sinon
    Ecrire ("Ce nombre est nul");
Finsi ;
Finsi ;
Fin.

```

4. Écrire un algorithme qui demande un réel à l'utilisateur et affiche sa valeur absolue (sans utiliser la fonction prédéfinie).

```

Algorithme Exo4
Var
X : réel ;
Début
Ecrire ("Ce programme affiche la valeur absolue d'un nombre");
Ecrire ("Entrez la valeur de ce nombre");
Lire (X);
    Si ( $X < 0$ ) Alors
         $X \leftarrow -1 * X$ ;
    Finsi ;
Ecrire ("La valeur absolue de ce nombre est : ", X);
Fin.

```

5. Écrire un algorithme qui affiche si un nombre entier saisi au clavier est pair ou impair.

```

Algorithme Exo5
Var
N : entier ;
Début
Ecrire ("Ce programme affiche si un nombre entier est pair ou impair");
Ecrire ("Entrez la valeur de ce nombre");

```

Lire (N) ;  
**Si** (N MOD 2 = 0) **Alors**  
    Ecrire ("Ce nombre est pair") ;  
**Sinon**  
    Ecrire ("Ce nombre est impair") ;  
**Finsi** ;  
**Fin.**

6. Écrire un algorithme qui teste si une note saisie au clavier est comprise entre 0 et 20.

Les conditions possibles :

note  $\geq 0$  et note  $\leq 20$

**Algorithme Exo6**

Var  
N : réel ;  
**Début**  
Ecrire ("Ce programme teste si une note saisie au clavier est comprise entre  
0 et 20") ;  
Ecrire ("Entrez une note") ;  
Lire (N) ;  
**Si** (note  $\geq 0$ ) et (note  $\leq 20$ ) **Alors**  
    Ecrire ("La note est comprise entre 0 et 20") ;  
**Finsi** ;  
**Fin.**

7. Écrire un algorithme qui permet d'afficher le maximum parmi deux nombres saisis au clavier.

**Algorithme Exo7**

Var  
N, M : entier ;  
**Début**

Ecrire ("Ce programme afficher le maximum parmi deux nombres saisis au clavier") ;

Ecrire ("Entrez la valeur du premier nombre") ;

Lire (N) ;

Ecrire ("Entrez la valeur du deuxième nombre") ;

Lire (M) ;

**Si** ( $N > M$ ) **Alors**

Ecrire ("Le maximum de ces deux nombres est : ", N) ;

**Sinon**

**Si** ( $M > N$ ) **Alors**

Ecrire ("Le maximum de ces deux nombres est : ", M) ;

**Sinon**

Ecrire ("Les deux nombres sont égaux") ;

**Finsi** ;

**Finsi** ;

**Fin.**

**Algorithme** Exo7

Var

N, M, Max : entier ;

**Début**

Ecrire ("Ce programme afficher le maximum parmi deux nombres saisis au clavier") ;

Ecrire ("Entrez la valeur du premier nombre") ;

Lire (N) ;

Ecrire ("Entrez la valeur du deuxième nombre") ;

Lire (M) ;

Max  $\leftarrow$  M ;

**Si** ( $N > M$ ) **Alors**

Max  $\leftarrow$  N ;

**Finsi** ;

Ecrire ("Le maximum de ces deux nombres est : ", Max) ;

**Fin.**

8. Écrire un algorithme qui demande le numéro d'un mois et affiche le nombre de jours que comporte ce mois.

**Algorithme Exo8**

Var

M : entier ;

**Début**

Ecrire ("Ce programme affiche le nombre de jours que comporte un mois  
saisie au clavier") ;

Ecrire ("Entrez un mois") ;

Lire (M) ;

**Si** (M = 2) **Alors**

Ecrire ("Le nombre de jours que comporte ce mois est 28 ou 29") ;

**Sinon**

**Si** (M = 4) ou (M = 6) ou (M = 9) ou (M = 11) **Alors**

Ecrire ("Le nombre de jours que comporte ce mois est 30") ;

**Sinon**

**Si** (M = 1) ou (M = 3) ou (M = 5) ou (M = 7) ou (M = 8) ou (M =  
10) ou (M = 12) **Alors**

Ecrire ("Le nombre de jours que comporte ce mois est 31") ;

**Sinon**

Ecrire ("Il n'existe pas un mois avec ce nombre") ;

**Finsi ;**

**Finsi ;**

**Finsi ;**

**Fin.**

**Algorithme Exo8**

Var

M : entier ;

**Début**

Ecrire ("Ce programme affiche le nombre de jours que comporte un mois saisi au clavier");

Ecrire ("Entrez un mois");

Lire (M);

**Selon M**

1 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

2 : Ecrire ("Le nombre de jours que comporte ce mois est 28 ou 29");

3 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

4 : Ecrire ("Le nombre de jours que comporte ce mois est 30");

5 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

6 : Ecrire ("Le nombre de jours que comporte ce mois est 30");

7 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

8 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

9 : Ecrire ("Le nombre de jours que comporte ce mois est 30");

10 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

11 : Ecrire ("Le nombre de jours que comporte ce mois est 30");

12 : Ecrire ("Le nombre de jours que comporte ce mois est 31");

Autre : Ecrire ("Il n'existe pas un mois avec ce nombre");

**Finselon ;**

**Fin.**

9. Écrire un algorithme permettant d'afficher le mois en toutes lettres selon son numéro saisi au clavier.

**Algorithme Exo9**

Var

M : entier ;

**Début**

Ecrire ("Ce programme affiche le mois en toutes lettres selon son numéro saisi au clavier");

Ecrire ("Entrez un mois");

Lire (M);

**Selon M**

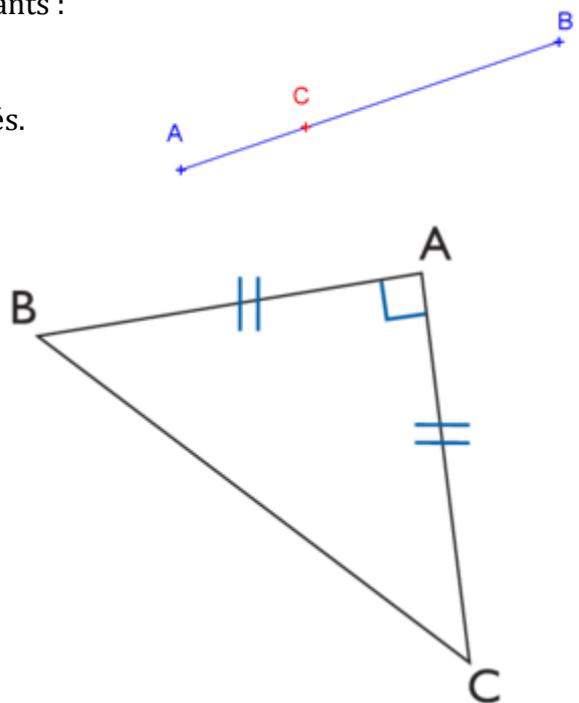
1 : Ecrire ("Janvier") ;  
 2 : Ecrire ("Février") ;  
 3 : Ecrire ("Mars") ;  
 4 : Ecrire ("Avril") ;  
 5 : Ecrire ("Mai") ;  
 6 : Ecrire ("Juin") ;  
 7 : Ecrire ("Juillet") ;  
 8 : Ecrire ("Aout") ;  
 9 : Ecrire ("Septembre") ;  
 10 : Ecrire ("Octobre") ;  
 11 : Ecrire ("Novembre") ;  
 12 : Ecrire ("Décembre") ;  
 Autre : Ecrire ("Il n'existe pas un mois avec ce nombre") ;  
**Finselon ;**

**Fin.**

10. Écrire un programme qui demande les coordonnées  $(x, y)$  des sommets A, B et C d'un triangle et affiche la nature du triangle (isocèle, équilatéral, rectangle ou quelconque).

L'algorithme doit faire apparaître les cas suivants :

- Triangle n'existe pas.
- Triangle aplati ou les points sont alignés.
- Triangle rectangle et isocèle en A.
- Triangle rectangle et isocèle en B.
- Triangle rectangle et isocèle en C.
- Triangle rectangle en A.
- Triangle rectangle en B.
- Triangle rectangle en C.
- Triangle isocèle en A.
- Triangle isocèle en B.
- Triangle isocèle en C.
- Triangle équilatéral.
- Triangle quelconque.



## Algorithme Exo1

Var

Ax, Bx, Cx, Ay, By, Cy : réel ;

AB, AC, BC : reel ;

### Début

Ecrire ("Ce programme affiche la nature d'un triangle") ;

Ecrire ("Entrez les coordonnées x, y du sommet A") ;

Lire (Ax, Ay) ;

Ecrire ("Entrez les coordonnées x, y du sommet B") ;

Lire (Bx, By) ;

Ecrire ("Entrez les coordonnées x, y du sommet C") ;

Lire (Cx, Cy) ;

$AB \leftarrow \sqrt{((Bx-Ax)*(Bx-Ax))+((By-Ay)*(By-Ay))}$  ;

$AC \leftarrow \sqrt{((Cx-Ax)*(Cx-Ax))+((Cy-Ay)*(Cy-Ay))}$  ;

$BC \leftarrow \sqrt{((Cx-Bx)*(Cx-Bx))+((Cy-By)*(Cy-By))}$  ;

**Si**  $(BC > AC + AB)$  ou  $(AC > BC + AB)$  ou  $(AB > BC + AC)$  **Alors**

Ecrire ("Le triangle n'existe pas") ;

### Sinon

**Si**  $(BC = AC + AB)$  ou  $(AC = BC + AB)$  ou  $(AB = BC + AC)$  **Alors**

Ecrire ("Les points sont alignés ou le triangle est aplati") ;

### Sinon

**Si**  $(BC * BC = AC * AC + AB * AB)$  et  $(AC * AC = AB * AB)$  **Alors**

Ecrire ("Le triangle est rectangle et isocèle en A") ;

### Sinon

**Si**  $(AC * AC = BC * BC + AB * AB)$  et  $(BC * BC = AB * AB)$  **Alors**

Ecrire ("Le triangle est rectangle et isocèle en B ") ;

### Sinon

**Si**  $(AB * AB = BC * BC + AC * AC)$  et  $(BC * BC = AC * AC)$  **Alors**

Ecrire ("Le triangle est rectangle et isocèle en C ") ;

### Sinon

**Si**  $(BC * BC = AC * AC + AB * AB)$  et  $(AC * AC \neq AB * AB)$  **Alors**

Ecrire ("Le triangle est rectangle en A ") ;

**Sinon**

**Si** ( $AC*AC=BC*BC+AB*AB$  et  $BC*BC \neq AB*AB$ ) **Alors**

Ecrire ("Le triangle est rectangle en B ");

**Sinon**

**Si** ( $AB*AB=BC*BC+AC*AC$  et  $BC*BC \neq AC*AC$ ) **Alors**

Ecrire ("Le triangle est rectangle en C");

**Sinon**

**Si** ( $AB*AB=AC*AC$  et  $BC*BC \neq AB*AB$ ) **Alors**

Ecrire ("Le triangle est isocèle en A");

**Sinon**

**Si** ( $BC*BC=AB*AB$  et  $BC*BC \neq AC*AC$ ) **Alors**

Ecrire ("Le triangle est isocèle en B");

**Sinon**

**Si** ( $BC*BC=AC*AC$  et  $BC*BC \neq AB*AB$ ) **Alors**

Ecrire ("Le triangle est isocèle en C");

**Sinon**

**Si** ( $AB*AB=AC*AC$  et  $BC*BC=AB*AB$ ) **Alors**

Ecrire ("Le triangle est équilatéral");

**Sinon**

Ecrire ("Le triangle est quelconque");

**Finsi ;**

**Fin.**

11. Écrire un algorithme qui vérifie si une année est bissextile. On rappelle qu'il y a des années bissextiles tous les 4 ans, mais la première année d'un siècle ne l'est pas (1800, 1900 n'étaient pas bissextiles) sauf tous les 400 ans (2000 était une année bissextile).

Les années bissextiles sont les années qui sont :

- Soit divisibles par 4 mais non divisibles par 100;
- Soit divisibles par 400.

**Algorithme Exo11**

Var

a : entier ;

b : booléen ;

**Début**

Ecrire ("Ce programme vérifie si une année est bissextile") ;

Ecrire ("Entrez l'année") ;

Lire (a) ;

Si  $((a \bmod 400) = 0)$  ou  $((a \bmod 4) = 0)$  et  $((a \bmod 100) \neq 0)$  **Alors**

    Ecrire (a, "est une année bissextile.") ;

**Sinon**

    Ecrire (a, "n'est pas une année bissextile.") ;

**Finsi ;**

**Fin.**

12. Écrire un algorithme qui demande une date sous la forme de 2 nombres entiers (numéro du jour et numéro du mois) et affiche la saison (ex : 12/02 ; hiver). On supposera que le premier jour de la saison est toujours le 21.

- L'hiver aura lieu du 21 décembre au 19 mars
- Le printemps aura lieu du 20 mars au 20 juin
- L'été aura lieu du 21 juin au 22 septembre
- L'automne aura lieu du 23 septembre au 20 décembre

13. Écrire l'algorithme d'un programme permettant de simuler le fonctionnement d'une calculatrice simple (+, -, \*, /). L'utilisateur saisira les deux opérandes, l'opérateur et le programme lui affichera le résultat correspondant. Dans le cas d'une division, on vérifiera bien que le dénominateur est non nul.

Algo exo 13

Var a, b : reel ;

    C : char ;

Debut

Ecrire (" donner les deux opérandes a calculer") ;

Lire (a,b) ;

Ecrire (" donner l'opération que vous voulez réaliser ") ;

Lire (c) ;

Selon c vaut

'+' : Ecrire (" le résultat est ", a+b) ;

'-' : Ecrire (" le résultat est ", a-b) ;

'\*' : Ecrire (" le résultat est ", a\*b) ;

'/' : si ( b ≠ 0 ) alors

    Ecrire (" le résultat est ", a/b) ;

    Sinon

    Ecrire (" division par zero ") ;

    Finsi ;

Autre : Ecrire (" opération n'existe pas") ;

Fin.

14. Écrire un algorithme permettant de calculer l'intersection de deux intervalles d'entiers [a, b] et [c, d] donnés (par exemple, l'intersection des intervalles [1,6] et [3,11] est l'intervalle [3,6], l'intersection des intervalles [1,6] et [9,11] est vide, l'intersection des intervalles [1,6] et [6,11] est l'intervalle [6,6]).

15. Écrire un algorithme permettant de calculer le montant de des allocations familiales sachant que le montant dépend du nombre d'enfant :
- a. Si nombre d'enfants est inférieur ou égale à trois alors les allocations sont de 200 Da par enfant.
  - b. Si le nombre d'enfants est strictement supérieur à trois et inférieurs ou égale à six alors les allocations sont de :
    - i. 200 Da par enfant pour les trois premiers enfants,
    - ii. 50 Da par enfant pour les suivants.
  - c. Si le nombre d'enfants est strictement supérieur à six alors les allocations sont de :
    - i. 200 Da par enfant pour les trois premiers enfants,
    - ii. 50 Da par enfant pour les 3 suivants,
    - iii. 0 Da par enfant pour les suivants.

Solution

Algo exo 15

Var n : entier ; m : réels ;

Debut

Ecrire (" donner le nombre des enfants ") ;

Lire (n) ;

Si (n <= 3 ) alors

M <- n\*200 ;

Sinon si (n > 3 et n <= 6 ) alors

M <- 600+ (6-n)\*50 ;

Sinon

M <- 750 ;

Finsi ;

Finsi ;

Ecrire (" le montant est ", M) ;

Fin.

النجاح في شيء يأتي من الاقتناع به