

Exercice N°01 : (Algorithme → Programme)

Soit l'algorithme suivant :

```
Algorithme ex01 ;
Variables
a, b, c, d: entier ;
Début
  {-*-*- Entrées -*-*-}
  Écrire('Donner trois nombres entiers : ');
  Lire(a,b,c);

  {-*-*- Traitements -*-*-}

  Si (a<b) alors
    Si (a<c) alors
      d ← a
    Sinon
      d ← c ;
  Fin-Si

  Sinon
    d ← c ;
  Fin-Si

  {-*-*- Sorties -*-*-}
  Écrire ('Le résultat = ', d) ;
Fin.
```

Questions :

- 1- Traduire l'algorithme en un programme en langage Pascal.
- 2- Compiler et exécuter le programme pour les valeurs suivantes (a=1, b=2, c=4) puis pour (a=4, b=2, c=0) puis pour (a=2, b= -1, c=4)
- 3- Que fait cet algorithme ?
- 4- Dérouler l'algorithme/Programme pour les cas suivants :
 - a=1, b=2, c=4
 - a=4, b=2, c=0

Exercice N°02 :

Ecrire un programme Pascal qui permet de lire deux nombres entiers (a, b), calcule et affiche la valeur de c comme suit :

$$c = \begin{cases} 0 & \text{si } a = b \\ a - b & \text{si } a > b \\ b - a & \text{si } a < b \end{cases}$$

Exercice N°03 :

Ecrire un programme Pascal intitulé **ordre_croissant**, qui permet d'afficher trois valeurs numérique A, B et C avec ordre croissant ?

Exercice N°04 :

Ecrire un programme Pascal intitulé **PARITE** qui saisit un nombre entier et détecte si ce nombre est pair ou impair.

Corrigé de l'exercice N°01 : (Algorithme → Programme)

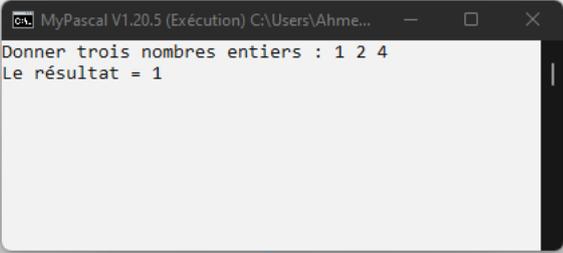
1) Traduire l'algorithme en Programme PASCAL

Algorithme	Programme PASCAL
<p>Algorithme exo1 ;</p> <p>Variables</p> <p>a, b, c, d : entier ;</p> <p>Début</p> <p>{-*-*- Entrées -*-*-}</p> <p>Écrire ('Donner trois nombres entiers : ');</p> <p>Lire (a,b,c) ;</p> <p>{-*-*- Traitements -*-*-}</p> <p>Si (a<b) alors</p> <p> Si (a<c) alors</p> <p> d ← a</p> <p> Sinon</p> <p> d ← c ;</p> <p> Fin-Si</p> <p>Sinon</p> <p> Si (b<c) alors</p> <p> d ← b</p> <p> Sinon</p> <p> d ← c ;</p> <p> Fin-Si</p> <p>Fin-si</p> <p>{-*-*- Sorties -*-*-}</p> <p>Écrire ('Le résultat = ', d) ;</p> <p>Fin.</p>	<p>Program exo1 ;</p> <p>Var</p> <p>a, b, c, d: integer ;</p> <p>Begin</p> <p>{-*-*- Entrées -*-*-}</p> <p>Write('Donner trois nombres entiers : ');</p> <p>Read(a,b,c) ;</p> <p>{-*-*- Traitements -*-*-}</p> <p>If (a<b) then</p> <p> If (a<c) then</p> <p> d:=a</p> <p> Else</p> <p> d:=c</p> <p>Else</p> <p> If (b<c) then</p> <p> d:=b</p> <p> Else</p> <p> d:=c ;</p> <p>{-*-*- Sorties -*-*-}</p> <p>Write ('Le résultat = ', d) ;</p> <p>End.</p>

2) Compiler et exécuter le programme pour les valeurs suivantes :

➤ a=1, b=2, c=4

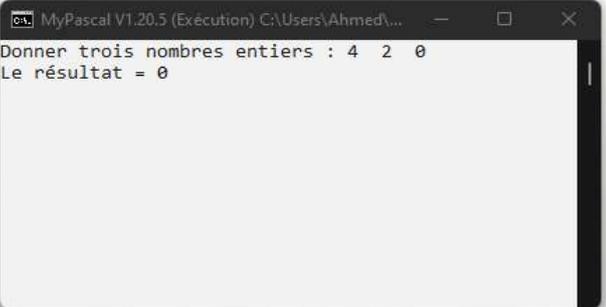
```
1 Program exo1 ;
2 Var
3   a, b, c, d: integer ;
4
5 Begin
6   {-*- Entrées -*-}
7   Write('Donner trois nombres entiers : ');
8   Read(a,b,c);
9
10  {-*- Traitements -*-}
11  if (a<b) then
12    if (a<c) then
13      d:=a
14    else
15      d:=c
16  else
17    if (b<c) then
18      d:=b
19    else
20      d:=c ;
21
22  {-*- Sorties -*-}
23  Write ('Le résultat = ', d);
24 End.
25
26
```



Après l'exécution

➤ a=4, b=2, c=0

```
1 Program exo1 ;
2 Var
3   a, b, c, d: integer ;
4
5 Begin
6   {-*- Entrées -*-}
7   Write('Donner trois nombres entiers : ');
8   Read(a,b,c);
9
10  {-*- Traitements -*-}
11  if (a<b) then
12    if (a<c) then
13      d:=a
14    else
15      d:=c
16  else
17    if (b<c) then
18      d:=b
19    else
20      d:=c ;
21
22  {-*- Sorties -*-}
23  Write ('Le résultat = ', d);
24 End.
25
26
```



Après l'exécution

➤ a=2, b= -1, c=4

```

1 Program exo1 ;
2 Var
3   a, b, c, d: integer ;
4
5 Begin
6   {*-**- Entrées *-**-}
7   Write('Donner trois nombres entiers : ');
8   Read(a,b,c);
9
10  {*-**- Traitements *-**-}
11  if (a<b) then
12    if (a<c) then
13      d:=a
14    else
15      d:=c
16  else
17    if (b<c) then
18      d:=b
19    else
20      d:=c;
21
22  {*-**- Sorties *-**-}
23  Write ('Le résultat = ', d);
24 End.
25
26

```

MyPascal V1.20.5 (Exécution) C:\Users\Ahm...
 Donner trois nombres entiers : 2 -1 4
 Le résultat = -1

Après l'exécution

3) Que fait cet algorithme ?

Le programme donne le plus petit des trois nombres entiers

4-1) Déroulement du programme pour a=1, b=2 et c=4

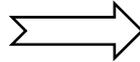
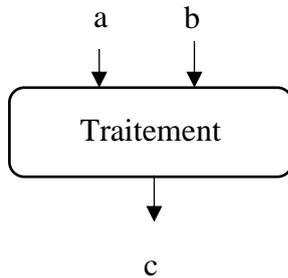
Instructions	Variables				Affichage
	a	b	c	d	
Écrire ('Donner trois nombres entiers : ');	/	/	/	/	Donner trois nombres entiers :
Lire (a,b,c) ;	1	2	4	/	
Si (a<b) alors 1 < 2 True ⇒ On exécute le bloc Si Si (a<c) alors 1 < 4 True ⇒ On exécute le bloc Si d ← a d ← 1 Fin-Si	1	2	4	1	
Écrire ('Le résultat = ', d) ;	1	6	4	1	Le résultat = 1

4-2) Déroulement du programme pour a=4, b=2 et c=0

Instructions	Variables				Affichage
	a	b	c	d	
Écrire ('Donner trois nombres entiers : ');	/	/	/	/	Donner trois nombres entiers :
Lire (a,b,c) ;	4	2	0	/	
<p>Si (a<b) alors 4 < 2 False ⇒ On n'exécute pas le bloc Si (on passe au bloc Sinon) Si (b<c) alors 4 < 0 False ⇒ On n'exécute pas le bloc Si (on passe au bloc Sinon) d ← c ; d ← 0 Fin-Si</p>	4	2	0	0	
Écrire ('Le résultat = ', d) ;	4	2	0	0	Le résultat = 0

Corrigé de l'exercice N°02 :

Le schéma entrées, traitement et sorties.



Entrées :

Lire (a,b) ;

Traitements :

```

Si a=b alors
  c ← 0;
Sinon-Si a>b alors
  c ← a-b;
Sinon
  c ← b-a;
Fin-Si
Fin-Si
  
```

Sortie :

Écrire (c) ;

Algorithme	Programme PASCAL
Algorithme exo2; Variables a, b, c : réel; Début {-*-*- Entrées -*-*-} Lire (a,b); {-*-*- Traitements -*-*-} Si a=b alors c ← 0; Sinon-Si a>b alors c ← a-b; Sinon c ← b-a; Fin-Si Fin-Si {-*-*- Sorties -*-*-} Écrire ('c = ', c:0:3) Fin.	program exo2; Var a, b, c : real; Begin {-*-*- Entrées -*-*-} Read (a,b); {-*-*- Traitements -*-*-} If a=b then c := 0 Else if a>b then c := a-b Else c := b-a; {-*-*- Sorties -*-*-} Write ('c = ', c:0:3) End.

Corrigé de l'exercice N°03 :

Algorithme	Programme PASCAL
<pre>Algorithme ordre_croissant; Variables A, B, C : entier; Début {-*-*- Entrées -*-*-} Écrire('Donner trois valeurs entière A, B et C : '); Read(A, B, C); {-*-*- Traitement & Sorties -*-*-} Si (A <= B) ET (B <= C) alors Écrire(A, B, C); Sinon Si (A <= C) ET (C <= B) alors Écrire (A, C, B); Sinon Si (B <= A) ET (A <= C) alors Écrire(B, A, C) Sinon Si (B <= C) ET (C<=A) alors Écrire(B, C, A) Sinon Si (C <= A) ET (A<=B) alors Écrire(C, A, B); Sinon Écrire(C, B, A); Fin-Si ; Fin-Si ; Fin-Si ; Fin-Si ; Fin-Si ; Fin.</pre>	<pre>Program ordre_croissant; Var A, B, C : integer; Begin {-*-*- Entrées -*-*-} Write('Donner trois valeurs entière A, B et C : '); Read(A, B, C); {-*-*- Traitement & Sorties -*-*-} Writeln('Les 3 valeurs avec un ordre croissant : '); if (A <= B) And (B <= C) then Write(A, B:5, C:5) else if (A <= C) And (C <= B) then Write(A, C:5, B:5) else if (B <= A) And (A <= C) then Write(B, A:5, C:5) else if (B <= C) And (C <= A) then Write(B, C:5, A:5) else if (C <= A) And (A <= B) then Write(C, A:5, B:5) else Write(C, B:5, A:5); End.</pre>

NB : Il existe d'autres solutions qui permettent d'afficher trois valeurs numériques A, B et C avec ordre croissant.

Corrigé de l'exercice N°04 :

Algorithme	Programme PASCAL
<p>Algorithme Partie ;</p> <p>Variables N : entier;</p> <p>Début</p> <p>{-*-*- Entrées -*-*-}</p> <p>Écrire ('Donner un entier : '); Lire (N);</p> <p>{-*-*- Traitement & Sorties-*-*-}</p> <p>Si N mod 2 = 0 alors Écrire (N, ' est pair')</p> <p>Sinon Écrire (N, ' est impair');</p> <p>Fin-Si ;</p> <p>Fin.</p>	<p>Program Parite ;</p> <p>Var N : Integer ;</p> <p>Begin</p> <p>{-*-*- Entrées -*-*-}</p> <p>Writeln ('Donner un entier : '); Readln (N);</p> <p>{-*-*- Traitement & Sorties-*-*-}</p> <p>IF N mod 2 = 0 Then Writeln (N, ' est pair')</p> <p>Else Writeln (N, ' est impair');</p> <p>End.</p>