

Série d'exercices n° 1
(Systèmes de Numération & Codage de l'Information)

Exercice 1: Parmi les nombres suivants, quels sont ceux qui peuvent être écrits en binaire ? En décimal ? En hexadécimal ?

	ABC	789	7B9	1000b	1	10200 ₂	GAFFE	0x789	1011	1000 ₁₀
Binaire										
Décimal										
Hexadécimal										

Exercice 2: Compléter le tableau suivant.

Binaire	Octal	Décimal	Hexadécimal	DCB à 8 bits
10100011				
	4560			
			D05	
				0000010000000111
			F4	
	1070			
		2011		
11111111				
			1EC	

Exercice 3: Voici des informations, tels que l'on pourrait les voir à l'intérieur de la mémoire d'un ordinateur :

```

001111000010101000100011011000110110010101100011
011010010010000100101000011001010111001101110100
001010010010011001110101011011100011111100101011
011011010110010101110011011100110110000101100111
011001010010000100101101011100110111010101100010
011011000110100101101101011010010110111001100001
01101100001000110010101000111110
```

Quelle signification accorder aux informations enregistrées ? La réponse dépend du codage qui a été employé pour les enregistrer.

Décoder le « message » ci-dessus en considérant successivement :

- Que sont codés des entiers naturels sur 8 bits,
- Que sont codés des entiers naturels sur 16 bits,
- Que sont codés des caractères, selon le code ASCII.

On se limitera au début du « message ».

Exercice 4 : Voici les codes ASCII, sur un octet chacun, de quelques caractères alphabétiques :

- 'A': 01000001; 'B': 01000010; 'C': 01000011
- 'a': 01100001; 'b': 01100010; 'c': 01100011

1. Ecrire les valeurs en base 10 correspondant à ces nombres binaires.
2. Quel doivent être les codes ASCII de D et de d ? De Z et de z ?

Table des Codes ASCII :

Code	Caractère	Code	Caractère	Code	Caractère	Code	Caractère
32	Space	33	!	34	"	35	#
36	\$	37	%	38	&	39	'
40	(41)	42	*	43	+
44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3
52	4	53	5	54	6	55	7
56	8	57	9	58	;	59	;
60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C
68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K
76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S
84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[
92	\	93]	94	^	95	-
96	'	97	a	98	b	99	c
100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k
108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s
116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{
124		125	}	126	~	127	Del

Exercice 1:

	ABC	789	7B9	1000b	1	10200 ₂	GAFFE	0x789	1011	1000 ₁₀
Binaire					✗	✗			✗	
Décimal		✗			✗				✗	✗
Hexadécimal	✗	✗	✗		✗				✗	

Exercice 2:

Binaire	Octal	Décimal	Hexadécimal	DCB à 8 bits
10100011	243	163	A3	00000001 00000110 00000011
100101110000	4560	2416	970	00000010 00000100 00000001 00000110
110100000101	6405	3333	D05	00000011 00000011 00000011 00000011
101111	57	47	2F	00000100 00000111
11110100	364	244	F4	00000010 00000100 00000100
1000111000	1070	568	238	00000101 00000110 00001000
11111011011	3733	2011	7DB	00000010 00000000 00000001 00000001
11111111	377	255	FF	00000010 00001001 00001001
111101100	754	492	1EC	00001000 00001001 00000010

Exercice 3:

On suppose que sont codés des entiers naturels sur 8 bits. Pour décoder, il suffit de regrouper les bits 8 par 8, et de calculer pour chacun des octets ainsi obtenus, le nombre qui y est codé : le premier octet 00111100 code l'entier 60 . . . Le message complet code la liste suivante d'entiers :

60 42 35 99 101 99 105 33 40 101 115 116 41 38 117 110 63 43 109 101 115
115 97 103 101 33 45 115 117 98 108 105 109 105 110 97 108 35 42 62

On suppose que sont codés des entiers naturels sur 16 bits. Le décodage est le même, mais en regroupant les bits 16 par 16. Les entiers codés par le message sont :

15402 9059 25955 26913 10341 29556 10534 30062 16171 28005 29555
24935 25889 11635 30050 27753 28009 28257 27683 10814

On suppose que sont codés des caractères. Pour décoder ce message, on peut utiliser le décodage en entiers 8 bits, puis consulter la table du code ASCII. On obtient alors le message suivant :

<#ceci!(est)&un?+message!-subliminal#>

Exercice 4:

1. Les valeurs en base 10 correspondants aux codes ASCII fournis sont les suivantes :
 - A : 65 ; B : 66 ; C : 67
 - a : 97 ; b : 98 ; c : 99
2. La suite des codes ASCII respectant l'ordre alphabétique, les codes manquant sont :
 - D : 01000100 (68 en décimal)
 - d : 01100100 (100 en décimal)
 - Z : 01011010 (90 en décimal)
 - z : 01111010 (122 en décimal)

Série d'exercices n° 1
(Généralités sur l'informatique)

Exercice 1: De quoi se sont constitués les éléments suivants :

- Unité centrale.
- Mémoire centrale.
- U. Arithmétique et Logique.
- U. d'entrée.
- U. de sortie.
- U. de traitement.
- U. de contrôle.

Exercice 2: Mettre une croix dans les cases adéquates :

est une mémoire	Permanente	Volatile	Secondaire
La RAM			
La ROM			
Disque dur			
Clé USB (flash disque)			

Exercice 3: Donner une définition et un exemple pour:

- Un périphérique d'entrée.
- Un périphérique de sortie.
- Un périphérique de stockage.

Exercice 4 : Classer les périphériques suivants selon leur type (entrée/ sortie/ stockage):

Scanner	Clavier	Lecteur de disquettes	Disque dur	Ecran
Haut-parleur	Lecteur DVD	Ecouteur	Imprimante	Microphone
CD / DVD	Clé USB (Flash disque)	Webcam	Souris	Onduleur

Exercice 5 :

Donner l'équivalent décimal des nombres : $(10010011)_2$, $(10010011)_{DCB4bits}$, $(734)_8$, $(21B)_{16}$.

Exercice 6 :

Donner l'équivalent binaire pur, DCB, Octal, hexadécimal des nombres décimaux (base 10) suivants : 0, 2, 8, 16, 734, 989.

Exercice 7 : Comparer les couples de nombres suivants :

$(11011000)_2$ et $(11001000)_2$, $(10001000)_2$ et $(10001001)_2$, $(11011000)_2$ et $(1101100)_2$,
 $(1100)_2$ et $(11)_{10}$, $(33)_8$ et $(33)_{10}$, $(23)_{16}$ et $(32)_8$, $(23)_7$ et $(24)_6$.

Exercice 8 :

Convertir en Octal, Hexadécimal et Décimal les nombres Binaires : 1100.001 ; 11.0101 ; 101.11.

Exercice 1:

- L'unité centrale est le cœur de l'ordinateur qui est actif en permanence (dès que l'ordinateur est allumé). Elle comporte : l'unité de traitement (CPU ou bien le processeur) et la mémoire centrale.
- La mémoire centrale : c'est là que sont mémorisés les informations et les programmes qui les traitent, tout au long de l'exécution. Elle est composée de deux mémoire RAM et ROM.
- L'unité centrale de traitement (CPU en anglais), ou processeur comporte deux parties l'unité de calcul "UAL" et l'unité de commande ou de contrôle "UC".
- L'unité de contrôle est un circuit électronique chargée de provoquer dans l'unité de calcul les opérations prévues par le programme. Elle contrôle aussi l'ensemble des autres dispositifs de l'ordinateur.
- Unité Arithmétique et Logique UAL est un circuit combinatoire regroupant différentes opérations arithmétiques (addition, soustraction, etc.) et logiques (AND, OR, XOR, etc.)
- L'unité d'entrée est un dispositif électronique permet d'envoyer des informations à un ordinateur. Par exemple : clavier, souris, lecteur de cartes, crayon optique, scanner...
- L'unité de sortie est un dispositif électronique permet d'extraire les informations de l'ordinateur. Par exemple : écran, imprimante, baffles...

Exercice 2:

est une mémoire	Permanente	Volatile	Secondaire
La RAM		X	
La ROM	X		
Disque dur	X		X
Clé USB (flash disque)	X		X

Exercice 3:

- **Un périphérique d'entrée** est un dispositif électronique permet d'envoyer des informations à un ordinateur. Par exemple : clavier, souris, lecteur de cartes, crayon optique, scanner...
- **Un périphérique de sortie** est un dispositif électronique permet d'extraire les informations de l'ordinateur. Par exemple : écran, imprimante, baffles....
- **Un périphérique de stockage** est un dispositif électronique permet de mémoriser d'une manière permanente les informations. exemple : disque dur, clé USB, disquette, CD-ROM ...

Exercice 4:

Périphérique d'entrée	Périphérique de sortie	Périphérique de stockage
Scanner	Ecran	Disque dur
Clavier	Haut-parleur	CD / DVD
Lecteur de disquettes	Lecteur de disquettes	Clé USB (Flash disque)
Lecteur DVD		
Microphone	Ecouteur	
Webcam	Imprimante	
Souris		

- **L'onduleur** : c'est un dispositif de protection de notre ordinateur. il régularise le courant électrique qui traverse notre ordinateur et possède les réserves d'énergie après coupure électrique.

Exercice 5 :

Nombre	Equivalent décimal
$(10010011)_2$	147
$(1001\ 0011)_{DCB4bits}$	93
$(734)_8$	476
$(21B)_{16}$	539

Exercice 6 :

Nombre Décimal	Binaire pur	DCB sur 4 bits	Octal	Hexadécimal
0	0	0000	0	0
2	10	0010	2	2
8	1000	1000	10	8
16	10000	0001 0110	20	10
734	1011011110	0111 0011 0100	1336	2DE
989	1111011101	1001 1000 1001	1735	3DD

Exercice 7 :

$(11011000)_2 > (11001000)_2$, $(10001000)_2 < (10001001)_2$, $(11011000)_2 > (1101100)_2$,

$(1100)_2 > (11)_{10}$, $(33)_8 < (33)_{10}$, $(23)_{16} > (32)_8$, $(23)_7 > (24)_6$

NB : Pour faire la comparaison quand il s'agit de deux bases différentes on devrait convertir vers une des deux bases ou bien convertir les deux bases vers la base décimale puis comparez. Si la comparaison des nombres binaire présente une difficulté vous pouvez convertir les deux nombres en décimal puis comparez.

Exercice 8 :

Nombre Binaire	Décimal	Octal	Hexadécimal
1100.001	12.125	$(\underline{001}\ \underline{100.001})_2 = (14,1)_8$	$(\underline{1100.0010})_2 = (C.2)_{16}$
11.0101	3.3125	$(\underline{011.010}\ \underline{100})_2 = (3,24)_8$	$(\underline{0011.0101})_2 = (3.5)_{16}$
101.11	5.75	$(\underline{101.110})_2 = (5,6)_8$	$(\underline{0101.1100})_2 = (5.C)_{16}$

Série d'exercices n° 2
(Algorithmes et Langages Algorithmiques)

Exercice 1:

Parmi les identificateurs suivants quels sont ceux qui sont correctement écrit et ceux qui ne le sont pas :

Xx25ab 2y fab & alpha 25,3 14.5 -11 c/n #ff a*b
 y² g! Ajou5 Ahmed gt-a "vap" F_i xy<5 d^y cinq google.fr

Exercice 2:

Compléter le tableau suivant par les constantes :

-0.5 1,8 '5' vap 'd' 0 +9 +10.6 "k=14.9" "15" "-11,6" `vap` "vap"

Entier	Réel	Caractère	Chaîne de caractère

Exercice 3:

Reprendre les expressions suivantes en algorithmique :

$$a \cdot x^2 + b \cdot x + c \qquad a - 3 \cdot c - \frac{a \cdot x}{(b - 1)^3} \qquad \frac{(-5x + \frac{-y}{3} - \frac{x^{1-y}}{(2-x)^4})^2}{8y}$$

Exercice 4:

Evaluer les expressions suivantes en respectant l'ordre des opérateurs et celle des parenthèses.

a-c/b-1 (a-1)/5-(b-c) Pour a=1, b=2, c=-1
 -1^5*x+y/(3+y) x↑-2↑(y/x/-2) Pour x=-3, y=1
 x↑(y/6)+3/y+y↑3/2*(y-1)↑(x-3) Pour x=4, y=3

Exercice 5:

Evaluer les expressions suivantes en respectant l'ordre des opérateurs et celle des parenthèses.

(x+y<4) et non(x<0) (x<y+3 ou y>3) et (x<1+y) x<y+3 ou y>3 et x<y+1
 Pour x=0, y=-1 puis pour x=-2, y=1

Exercice 6:

Evaluer les expressions suivantes.

p et non p et p ou non p p ou non p et non p p ou non p ou non p
 Pour p=Vrai puis p=Faux.

Exercice 1: un identificateur commence par une lettre alphabétique (majuscule ou bien minuscule) et il est composé uniquement de lettres alphabétiques ou chiffres

Identificateurs	Réponses
Xx25ab	Identifiant correct
2y	Identifiant incorrect car il commence par un chiffre au lieu une lettre
Fab	Identifiant correct
&	Identifiant incorrect car c'est un caractère spécial
Alpha	Identifiant correct
25,3	Identifiant incorrect car il commence par un chiffre et contient le caractère ','
14.5	Identifiant incorrect car il commence par un chiffre et contient le caractère '.'
-11	Identifiant incorrect car il commence par le caractère '-'
c/n	Identifiant incorrect car il contient le caractère '/'
#ff	Identifiant incorrect car il commence par le caractère '#'
a*b	Identifiant incorrect car il contient le caractère '*'
y²	Identifiant incorrect car un identificateur ne contient pas de puissance
g!	Identifiant incorrect car il contient le caractère ' !'
Ajou5	Identifiant correct
Ahmed	Identifiant correct
gt-a	Identifiant incorrect car il contient le caractère '-'
"vap"	Identifiant incorrect car il commence par le caractère '"'
F_i	Identifiant incorrect car un identificateur ne contient pas d'indice
xy<5	Identifiant incorrect car il contient le caractère '<'
d^y	Identifiant incorrect car un identificateur ne contient pas de puissance
Cinq	Identifiant correct
google.fr	Identifiant incorrect car il commence par le caractère '.'

Exercice 2:

Entier	Réel	Caractère	Chaine de caractères	Rien
0	-0.5	`5`	"k=14.9"	1,8
+9	+10.6	`d`	"15"	vap
			"-11,6"	`vap`
			"vap"	

Exercice 3:

Expression	Ecriture algorithmique
$a \cdot x^2 + b \cdot x + c$	$a*x \uparrow 2 + b*x + c$
$a - 3 \cdot c - \frac{a \cdot x}{(b - 1)^3}$	$a - 3*c - (a*x) / (b-1) \uparrow 3$
$\frac{(-5x + \frac{-y}{3} - \frac{x^{1-y}}{(2-x)^4})^2}{8y}$	$(-5*x + (-y) / 3 - (x \uparrow (1-y) / (2-x) \uparrow 4) \uparrow 2) / (8*y)$

NB : J'ai opté pour l'ordre de priorité suivant :

1. Parenthèses
2. Puissance (^ ou bien \uparrow)
3. Signe plus (+) et signe moins (-) unaires, négation logique (non)
4. Multiplication (*), division (/)
5. Addition (+) et soustraction (-)
6. Opérateurs de comparaison (<, <=, >, >=, =, <>)
7. Opérateur logique ET
8. Opérateur logique OU

L'évaluation est de gauche vers la droite pour les opérateurs de même type (associativité à gauche) même pour l'opérateur puissance « \uparrow ».

Exercice 4:

Expression	Paramètres	Evaluation
$a-c/b-1$	$a=1, b=2, c=-1$	$= (a-(c/b))-1$ $= (1-(-1/2))-1$ $= (1+0.5)-1$ $= 1.5-1$ $= 0.5$
$(a-1)/5-(b-c)$	$a=1, b=2, c=-1$	$= ((a-1)/5) - (b-c)$ $= ((1-1)/5) - (2-(-1))$ $= (0/5) - (2+1)$ $= 0-3$ $= -3$
$-1^5*x+y/(3+y)$	$x=-3, y=1$	$= (((-1)^5)*x) + (y / (3+y))$ $= ((-1)*(-3)) + (1 / (3+1))$ $= (3) + (1 / 4)$ $= 3+0.25$ $= 3.25$

$x \uparrow -2 \uparrow (y/x - 2)$	$x = -3, y = 1$	$= (x \uparrow (-2)) \uparrow ((y/x) / (-2))$ $= ((-3) \uparrow (-2)) \uparrow ((1 / (-3)) / (-2))$ $= (0.11) \uparrow (-0.33/2)$ $= (0.11) \uparrow (0.16)$ $= 0.69$
$x \uparrow (y/6) + 3/y + y \uparrow 3/2 * (y-1) \uparrow (x-3)$	$x = 4, y = 3$	$= (x \uparrow (y/6)) + (3/y) + ((y \uparrow 3)/2) * ((y-1) \uparrow (x-3))$ $= (4 \uparrow (3/6)) + (3/3) + ((3 \uparrow 3)/2) * ((3-1) \uparrow (4-3))$ $= (4 \uparrow (0.5)) + (1) + ((27)/2) * ((2) \uparrow (1))$ $= (2) + 1 + (13.5) * (2)$ $= 2 + 1 + 27$ $= (2 + 1) + 27$ $= 3 + 27$ $= 30$

Exercice 5:

Expression	Paramètres	Evaluation
$(x+y < 4)$ et $\text{non}(x < 0)$	$x=0, y=-1$	$= (0 + -1 < 4)$ et $(\text{non}(0 < 0))$ $= (-1 < 4)$ et $(\text{non}(\text{Faux}))$ $= (\text{Vrai})$ et (Vrai) $= \text{Vrai}$
	$x=-2, y=1$	$= (-2 + 1 < 4)$ et $(\text{non}(-2 < 0))$ $= (-1 < 4)$ et $(\text{non}(\text{Vrai}))$ $= (\text{Vrai})$ et (Faux) $= \text{Faux}$
$(x < y + 3$ ou $y > 3)$ et $(x < 1 + y)$	$x=0, y=-1$	$= ((0 < -1 + 3)$ ou $(-1 > 3))$ et $(0 < 1 + -1)$ $= ((0 < 2)$ ou $(-1 > 3))$ et $(0 < 0)$ $= (\text{Vrai}$ ou $\text{Faux})$ et (Faux) $= (\text{Vrai})$ et (Faux) $= \text{Faux}$
	$x=-2, y=1$	$= ((-2 < 1 + 3)$ ou $(1 > 3))$ et $(-2 < 1 + 1)$ $= ((-2 < 4)$ ou $(1 > 3))$ et $(-2 < 2)$ $= (\text{Vrai}$ ou $\text{Faux})$ et (Vrai) $= (\text{Vrai})$ et (Vrai) $= \text{Vrai}$
$x < y + 3$ ou $y > 3$ et $x < y + 1$	$x=0, y=-1$	$= ((x < y + 3)$ ou $(y > 3$ et $x < y + 1))$ $= ((0 < -1 + 3)$ ou $(-1 > 3$ et $0 < -1 + 1))$ $= ((0 < 2)$ ou $(-1 > 3$ et $0 < 0))$ $= (\text{Vrai}$ ou $(\text{Faux}$ et $\text{Faux}))$ $= (\text{Vrai}$ ou $\text{Faux})$ $= \text{Vrai}$
	$x=-2, y=1$	$= (x < y + 3$ ou $(y > 3$ et $x < y + 1))$ $= (-2 < 1 + 3$ ou $(1 > 3$ et $-2 < 1 + 1))$ $= (-2 < 4$ ou $(1 > 3$ et $-2 < 2))$ $= (\text{Vrai}$ ou $(\text{Faux}$ et $\text{Vrai}))$ $= (\text{Vrai}$ ou $\text{Faux})$ $= \text{Vrai}$

Exercice 6:

Expression	Paramètres	Evaluation
p et non p et p ou non p	p=Vrai	$= (((p \text{ et } (\text{non } p)) \text{ et } p) \text{ ou } (\text{non } p))$ $= (((\text{Vrai et } (\text{non Vrai})) \text{ et Vrai) ou (non Vrai))$ $= (((\text{Vrai et Faux}) \text{ et Vrai) ou (Faux))$ $= ((\text{Faux et Vrai}) \text{ ou } (\text{Faux}))$ $= (\text{faux}) \text{ ou } (\text{Faux})$ $= \text{Faux}$
	p=Faux	$= (((p \text{ et } (\text{non } p)) \text{ et } p) \text{ ou } (\text{non } p))$ $= (((\text{Faux et } (\text{non Faux})) \text{ et Faux) ou (non Faux))$ $= (((\text{Faux et Vrai}) \text{ et Faux) ou (Vrai))$ $= ((\text{Faux et Faux}) \text{ ou } (\text{Vrai}))$ $= (\text{faux}) \text{ ou } (\text{Vrai})$ $= \text{Vrai}$
p ou non p et non p	p=Vrai	$= (p \text{ ou } ((\text{non } p) \text{ et } (\text{non } p)))$ $= (\text{Vrai ou } ((\text{non Vrai}) \text{ et } (\text{non Vrai})))$ $= (\text{Vrai ou } ((\text{Faux}) \text{ et } (\text{Faux})))$ $= (\text{Vrai ou } (\text{Faux}))$ $= \text{Vrai}$
	p=Faux	$= (p \text{ ou } ((\text{non } p) \text{ et } (\text{non } p)))$ $= (\text{Faux ou } ((\text{non Faux}) \text{ et } (\text{non Faux})))$ $= (\text{Faux ou } ((\text{Vrai}) \text{ et } (\text{Vrai})))$ $= (\text{Faux ou } (\text{Vrai}))$ $= \text{Vrai}$
p ou non p ou non p	p=Vrai	$= ((p \text{ ou } (\text{non } p)) \text{ ou } (\text{non } p))$ $= ((\text{Vrai ou } (\text{non Vrai})) \text{ ou } (\text{non Vrai}))$ $= ((\text{Vrai ou } (\text{Faux})) \text{ ou } (\text{faux}))$ $= ((\text{Vrai}) \text{ ou } (\text{faux}))$ $= \text{Vrai}$
	p=Faux	$= ((p \text{ ou } (\text{non } p)) \text{ ou } (\text{non } p))$ $= ((\text{Faux ou } (\text{non Faux})) \text{ ou } (\text{non Faux}))$ $= ((\text{Faux ou } (\text{Vrai})) \text{ ou } (\text{Vrai}))$ $= ((\text{Vrai}) \text{ ou } (\text{Vrai}))$ $= \text{Vrai}$

Série d'exercices n° 3

Exercice 1:

1. Eliminer les variables, les actions inutiles et corriger les éventuelles erreurs pour les trois algorithmes présentés ci-dessous.

Algorithme Algo1 ; Constantes N=5; M : 3; (M=3;) Variables X,Y,Z : Réel ; E,D : Entier ; A,B : Booléen ; C : Caractère ; Début X ← N ² ; (X ← N ² *1.0 ;) Ecrire X ; (Ecrire(X) ;) Y ← X (Y ← X ;) Z ← M (Z ← M*1.0;) Ecrire(Y ; Z) ; (Ecrire(Y , Z) ;) E ← N X M ; (E ← N * M ;) D ← E MOD 2 ; Ecrire(D,E) ; Fin (Fin.)	Algorithme Algo2 ; Variables A,B,C : Booléen ; Début A= Vrai ; (A ← Vrai ;) Ecrire(A) ; B ← A ; Ecrire(B) ; A ← Faux ; B ← Non A ; Ecrire(B) ; C ← A et B ; Ecrire(B) ; (Ecrire(C) ;) C ← Non (A ou B) ; Ecrire(C) (Ecrire(C) ;) Fin (Fin.)	Algorithme Algo3 ; Variable a,b,c : chaîne de caractères ; Début a ← '423' ; (a ← "423" ;) b ← "abc.3" ; c = a & b ; (c ← a & b ;) Ecrire(c) ; Ecrire(b) ; Fin (Fin.)
--	---	---

2. Quel sera l'affichage après exécution des algorithmes corrigés.

Algorithme 1 :

Actions	N	M	X	Y	Z	E	D
Déclaration	5	3	?	?	?	?	?
X ← N ² *1.0 ;	5	3	25.0	?	?	?	?
Y ← X ;	5	3	25.0	25.0	?	?	?
Z ← M*1.0;	5	3	25.0	25.0	3.0	?	?
E ← N * M ;	5	3	25.0	25.0	3.0	15	?
D ← E MOD 2 ;	5	3	25.0	25.0	3.0	15	1

<u>Affichage :</u>		
25.0		(Ecrire(X) ;)
25.0	3.0	(Ecrire(Y , Z) ;)
1	15	(Ecrire(D,E) ;)

Algorithme 2 :

Actions	A	B	C
Déclaration	?	?	?
A ← Vrai ;	Vrai	?	?
B ← A ;	Vrai	Vrai	?
A ← Faux ;	Faux	Vrai	?
B ← Non A ;	Faux	Vrai	?
C ← A et B ;	Faux	Vrai	Faux
C ← Non (A ou B)	Faux	Vrai	Faux

<u>Affichage :</u>	
Vrai	(Ecrire(A) ;)
Vrai	(Ecrire(B) ;)
Vrai	(Ecrire(B) ;)
Faux	(Ecrire(C) ;)
Faux	(Ecrire(C) ;)

Algorithme 3 :

Actions	a	b	c
Déclaration	?	?	?
a ← "423 " ;	"423"	?	?
b ← "abc.3" ;	"423"	"abc.3"	?
c ← a & b ;	"423"	"abc.3"	"423abc.3"

Affichage :

423abc.3 (Ecrire(c) ;)
abc.3 (Ecrire(b) ;)

Exercice 2 :

Ecrire un Algorithme qui lit les informations d'un étudiant : Nom, Prénom, Age, Section d'étude, l'adresse, puis afficher la sortie comme suit :

Je m'appelle *Badis zakaria*

J'ai *20* ans

Section d'études : *K*

J'habite à : *5, Rue Mouloud Feraoun Blida*

Solution :

Algorithme exercice2 ;

Variables nom, prenom, adresse : **chaîne de caractères** ;

age : **entier** ;

section : **caractère** ;

Début

Écrire ("Donner le nom, le prénom, l'adresse, l'âge et la section") ;

Lire (nom, prenom, adresse, age, section) ;

Écrire ("*****") ;

Écrire ("Je m'appelle ", nom, " ", prenom) ;

Écrire ("J'ai ", age, "ans") ;

Écrire ("Section d'études : ", section) ;

Écrire ("J'habite à : ", adresse) ;

Écrire ("*****") ;

Fin.

Exercice 3:

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

Algorithme exercice2 ;

Variables A, B : entier ;

Début

A ← 5 ;

B ← 2 ;

A ← B ;

B ← A ;

Fin.

Moralité : les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

Solution :

a- Les valeurs des variables A et B après exécution :

Actions	A	B
Déclarations	?	?
$A \leftarrow 5$;	5	?
$B \leftarrow 2$;	5	2
$A \leftarrow B$;	2	2
$B \leftarrow A$;	2	2

b- Les deux dernières instructions ne permettent donc pas d'échanger les deux valeurs de B et A, puisque l'une des deux valeurs (celle de A) est ici écrasée. Si l'on inverse les deux dernières instructions, cela ne changera rien du tout, hormis le fait que cette fois c'est la valeur de B qui sera écrasée.

Exercice 4 :

Ecrire les algorithmes qui permettent de :

1-Calculer le produit de 3 nombres réels lus au clavier ainsi que leur somme. Afficher sous le format : le produit de ...,...,... est : ...

la somme de ...,...,... est : ...

2-Calculer la valeur de deux nombres entiers connaissant leur somme et leur différence.

3-Ecrire l'algorithme permettant de saisir l'abscisse d'un point A et de calculer son ordonné

$f(x) = 2x^3 - 3x^2 + 4$ Evaluer le résultat en expliquant les ordres de priorité pour $x = -2$.

Solution :

1.

```
Algorithme exercice41 ;
Variables X, Y, Z, prod, som : Réel ;
Début
Écrire ("Donner la valeur de X, Y et Z") ;
Lire (X, Y) ;
prod ← X*Y*Z ;
som ← X+Y+Z ;
Écrire ("Le produit de", X, ",", Y, ",", Z, " est : ", prod) ;
Écrire ("La somme de", X, ",", Y, ",", Z, " est : ", som) ;
Fin.
```

2.

```
Algorithme exercice42 ;
Variables A, B, D, S : Entier ;
Début
Écrire ("Donner la somme S ") ;
Lire (S);
Écrire ("Donner la différence D ") ;
Lire (D);
A ← (S+D) div 2;
B ← (S-D) div 2 ;
Écrire ("La valeur de A=", A) ;
Écrire ("La valeur de B=", B) ;
Fin.
```

3. a- L'algorithme:

```
Algorithme exercice43 ;
Variables X, Y: Réel ;
Début
Écrire ("Donner la valeur de l'abscisse X") ;
Lire (X) ;
 $Y \leftarrow 2 * X^3 - 3 * X^2 + 4$  ;
Écrire ("La valeur de l'ordonnée Y=", Y) ;
Fin.
```

3. b- Evaluation pour $x \leftarrow -2$:

```
 $Y \leftarrow 2 * (-2)^3 - 3 * (-2)^2 + 4$  ;
 $Y \leftarrow 2 * (-8) - 3 * (4) + 4$  ;
 $Y \leftarrow -16 - 12 + 4$  ;
 $Y \leftarrow -28 + 4$  ;
 $Y \leftarrow -24$  ;
```

Exercice 5:

1-Ecrire l'algorithme qui permet de permuter les valeurs de deux variables entières A et B.

a - En utilisant une variable auxiliaire.

b - Sans utiliser une variable auxiliaire.

2-Trouver un algorithme qui réalise la permutation circulaire de trois variables entières A, B et C en utilisant une autre variable D.

Exemple :

A	B	C	Permutation circulaire	A	B	C
10	5	-7		-7	10	5

Solution :

1a. Permutation des deux variables en utilisant une variable auxiliaire

```
Algorithme exercice51a ;
Variables A, B, C: Entier ;
Début
Écrire ("Donner la valeur de A et B") ;
Lire (A, B) ;
 $C \leftarrow A$  ;
 $A \leftarrow B$  ;
 $B \leftarrow C$  ;
Écrire (" A et B après permutation sont ", A, B) ;
Fin.
```

1b. Permutation des deux variables sans utilisation d'une variable auxiliaire

```
Algorithme exercice51b ;
Variables A, B: Entier ;
Début
Écrire ("Donner la valeur de A et B") ;
Lire (A, B) ;
 $A \leftarrow A+B$  ;
 $B \leftarrow A-B$  ;
 $A \leftarrow A-B$  ;
Écrire (" A et B après permutation sont ", A, B) ;
Fin.
```

2. Permutation circulaire :

```
Algorithme exercice52 ;  
Variables A, B, C, D: Entier ;  
Début  
  Écrire ("Donner la valeur de A, B et C") ;  
  Lire (A, B, C) ;  
  D ← C ;  
  C ← B ;  
  B ← A ;  
  A ← C ;  
  Écrire (" A, B, C après permutation sont ", A, B, C) ;  
Fin.
```

Exercice 6:

Donner l'algorithme qui permet le décodage en jour, mois et année d'une date stockée sous forme d'un entier de 6 chiffres JJMMAA.

Exemple : 270993 ==> jour : 27, mois : 09, année : 93

Solution :

```
Algorithme exercice6 ;  
Variables Date, JJ, MM, AA: Entier ;  
Début  
  Écrire ("Donner la valeur de la date") ;  
  Lire (date) ;  
  AA ← date div 10000 ;  
  date ← date mod 10000 ;  
  MM ← date div 100 ;  
  JJ ← date mod 100 ;  
  Si (JJ < 10) alors Écrire ("Le Jour est :0",JJ) ;  
    Sinon Écrire ("Le Jour est :",JJ) ;  
  Fsi ;  
  Si (MM < 10) alors Écrire ("Le Mois est :0",MM) ;  
    Sinon Écrire ("Le Mois est :",MM) ;  
  Fsi ;  
  Si (AA < 10) alors Écrire ("L'année est :0",AA) ;  
    Sinon Écrire ("L'année est :",AA) ;  
  Fsi ;  
Fin.
```

Exercice 7:

Lors d'un examen, un candidat a 4 notes d'écrit et 2 d'oral de coefficients respectifs : 4, 3, 1, 1 et 2, 1. On veut afficher, pour le candidat :

La note moyenne à l'écrit,

La note moyenne à l'oral,

La moyenne générale.

Écrire l'algorithme qui effectue le travail précédent, sachant que les notes sont lues à partir de clavier.

Solution :

```
Algorithme exercice7 ;
Variables ne1, ne2, ne3, ne4, no1, no2, me, mo, mg: Réel ;
Début
Écrire ("Donner les notes de l'écrit") ;
Lire (ne1, ne2, ne3, ne4) ;
me ← (ne1*4+ne2*3+ne3+ne4)/9 ;
Écrire ("Donner les notes de l'oral") ;
Lire (ne1, ne2, ne3, ne4) ;
mo ← (no1*2+no2*3)/3 ;
mg ← (me+mo)/2 ;
Écrire ("La note moyenne à l'écrit :", me) ;
Écrire ("La note moyenne à l'oral :", mo) ;
Écrire ("La moyenne générale :", mg)
Fin.
```

A remettre comme devoir

Exercice 8:

Dans un grimoire on trouve la conversation suivante :

"Quel âge avez-vous donc, noble vieillard ? J'ai deux fois votre âge, jeune homme, et quand j'avais l'âge que vous avez le roi Henri IV tombait sous le pont neuf"

Écrire un algorithme qui permet de déterminer l'âge du vieillard et celui du jeune homme.

Solution :

```
Algorithme exercice8 ;
Variables DateC, AgeJ, AgeV: Entier ;
Début
Écrire ("Donner la date de conversation") ;
Lire (DateC) ;
AgeJ ← DateC-1610 ;
AgeV ← AgeJ*2 ;
Écrire (" L'âge du noble vieillard ", AgeV) ;
Écrire (" L'âge du jeune", AgeJ) ;
Fin.
```

Exercice 9:

Écrire un algorithme qui lit :

Le prix d'un article,

Le nombre d'articles,

Le taux de TVA,

Et qui affiche comme résultat le prix total TTC des articles.

Solution :

Algorithme exercice9 ;

Variables PRIXA, TVA, TTC: **Réel** ;
NOMBREA : **Entier** ;

Début

Écrire ("Donner le nombre d'article NOMBREA") ;

Lire (NOMBREA) ;

Écrire ("Donner le prix d'un article PRIXA") ;

Lire (PRIXA) ;

Écrire ("Donner le taux de TVA) ;

Lire (TVA) ;

$TTC \leftarrow NOMBREA * PRIXA * (1 + TVA / 100)$;

Écrire ("Le Prix total des articles TTC= :",TTC)

Fin.

Corrigé de série d'exercices n° 4

Exercice 1:

Conditions	Instructions exécutées
Cond1 =Vrai Cond2 =Vrai Cond3 =Vrai	1, 2, 5
Cond1 =Faux Cond2 =Vrai Cond3 =Vrai	6,7
Cond1 =Faux Cond2 =Faux Cond3 =Faux	8

Exercice 2:

```
Algorithme Exo2;  
Variables a, b : entier ;  
Début  
Ecrire ("Donner un entier a") ;  
Lire(a) ;  
Si (a<0)  
Alors b←-a ;  
Sinon b←a ;  
Fsi ;  
Si (b MOD 2=0)  
Alors Ecrire ("La valeur absolue de ", a, " est : ", b, " et c'est un nombre pair") ;  
Sinon Ecrire ("La valeur absolue de ", a, " est : ", b, " et c'est un nombre impair") ;  
Fsi ;  
Fin.
```

Exercice 3:

```
Algorithme Exo3;  
Variables a, b : réel ;  
Début  
Ecrire ("Donner les valeurs de a et b") ;  
Lire(a, b) ;  
Si(a=0) ou (b=0)  
Alors Ecrire("Le produit est nul") ;  
Sinon Si ((a<0) et (b<0)) ou((a>0) et (b>0))  
Alors Ecrire("Le produit est positif") ;  
Sinon Ecrire("Le produit est négatif") ;  
Fsi ;  
Fsi ;  
Fin.
```

Exercice 4:

```
Algorithme Exo4;
Variables x, y : réel ;
Début
Ecrire ("Donner la valeur de x") ;
Lire(x) ;
Si(x<-1)
alors y←(x-1)^3 ;
Sinon Si(x≥0)
    alors y←(-3*x+7)/5 ;
    Sinon y←x^4+8;
    Fsi ;
Fsi ;
Ecrire("y=",y) ;
Fin.
```

Exercice 5:

```
Algorithme Exo5;
Variables a, b : réel ;
        op : caractère ;
Début
Ecrire ("donner les 2 opérandes") ;
Lire (a, b) ;
Ecrire("donner l'opérateur parmi : +, -, *, /");
Lire(op) ;
Selon op
    '+' : Ecrire ("La somme est", a+b);
    '-' : Ecrire ("La différence est", a-b);
    '*' : Ecrire ("Le produit est", a*b);
    '/' : Si (b ≠ 0)
        Alors Ecrire ("La division est", a/b);
        Sinon Ecrire ("Erreur : Division par zéro");
        Fsi ;
    Autres : Ecrire ("Erreur : Ce n'est pas un opérateur arithmétique") ;
FinSelon ;
Fin.
```

Exercice 6:

```
Algorithme Exo6;
Variables a, b, c, delta : réel ;
Début
Ecrire ("donner les valeurs de a, b, c") ;
Lire (a, b, c) ;
Si (a=0) et (b=0) et (c=0)
Alors Ecrire ("La solution est R") ;
Sinon Si (a=0) et (b=0)
    Alors Ecrire ("Pas de solutions") ;
Sinon Si (a=0)
    Alors Ecrire ("La solution est x=", -c/b) ;
    Sinon
Début
```

```

delta ← b2 - 4*a*c;
Si (delta < 0)
Alors Ecrire (" Pas de solutions dans R");
Sinon Si (delta = 0)
    Alors Ecrire ("Solution double x0=", -b/(2*a)) ;
    Sinon Ecrire ("Solutions différentes x1=", (-b + (delta)0.5)/(2*a),
" et x2=", (-b - (delta)0.5)/(2*a)) ;
    Fsi ;
Fsi ;
Fin ;
Fsi ;
Fsi ;

```

Trace d'exécution de cet algorithme avec l'équation $2X^2 + X - 1 = 0$

Actions	a	b	c	delta
Déclarations	?	?	?	?
Lire (a, b, c) ;	2	1	-1	?
delta ← b ² - 4* a*c ;	2	1	-1	9
Solutions différentes x ₁ =0.5 et x ₂ =-1				

Exercice 7:

```

Algorithme Exo7;
Variables a, b, c: réel ;
Début
{On suppose que les valeurs sont différentes}
Ecrire ("donner les valeurs de a, b, c") ;
Lire (a, b, c) ;
Si (a < b) et (a < c)
Alors Si (b < c)
    Alors Ecrire ("L'ordre croissant est :", a, b, c) ;
    Sinon Ecrire ("L'ordre croissant est :", a, c, b) ;
    Fsi ;
Sinon Si (b < a) et (b < c)
    Alors Si (a < c)
        Alors Ecrire ("L'ordre croissant est :", b, a, c) ;
        Sinon Ecrire ("L'ordre croissant est :", b, c, a) ;
        Fsi ;
    Sinon Si (a < b)
        Alors Ecrire ("L'ordre croissant est :", c, a, b) ;
        Sinon Ecrire ("L'ordre croissant est :", c, b, a) ;
        Fsi ;
    Fsi ;
Fsi ;
Fin.

```

Exercice 8:

```
Algorithme Exo8;
Variables H, M, S: Entier ;
Début
Ecrire ("Donner l'heure actuelle en Heure, Minute et Seconde") ;
Lire (H, M, S) ;
Si (H≥0) et (H≤23) et (M≥0) et (M≤59) et (S≥0) et (S≤59)
Alors
Début
S← S+ 30 ;
M← M+ S div 60 ;
S← S mod 60 ;
H← (H+ M div 60) mod 24 ;
M← M mod 60 ;
Ecrire (" L'heure après 30 secondes est ", H, " :", M, " :", S) ;
Fin ;
Sinon Ecrire (" L'heure n'est pas correcte ") ;
Fsi ;
Fin.
```

Exercice 9:

- La somme des n premiers nombres entiers positifs :

```
Algorithme Exo9;
Variables n, i, s: Entier ;
Début
Répéter
Ecrire ("Donner n>0") ;
Lire (n) ;
Jusqu'à (n>0) ;
S←0 ;
Pour i allant de 1 à n pas 1
Faire
s← s+ i ;
Fait ;
Ecrire ("La somme des ", n, "Premiers entiers est ", s) ;
Fin.
```

- La somme des carrés des n premiers nombres :

```
Algorithme Exo9;
Variables n, i, s: Entier ;
Début
Répéter
Ecrire ("Donner n>0") ;
Lire (n) ;
Jusqu'à (n>0) ;
S←0 ;
Pour i allant de 1 à n pas 1
Faire
s← s+ i↑2 ;
Fait ;
Ecrire ("La somme des carrés des ", n, "Premiers entiers est ", s) ;
Fin.
```

Exercice 10:

- Factoriel n par la méthode ascendante:

```
Algorithme Exo10a;  
Variables n, i, fact: Entier ;  
Début  
Répéter  
Ecrire ("Donner n≥0") ;  
Lire (n) ;  
Jusqu'à (n≥0) ;  
Si (n=0) ou (n=1)  
Alors Ecrire (n, " !=1") ;  
Sinon  
Début  
fact←1 ;  
Pour i allant de 2 à n pas 1  
Faire  
fact← fact*i ;  
Fait ;  
Ecrire (n, " !=", fact) ;  
Fin ;  
Fsi ;  
Fin.
```

- Factoriel n par la méthode descendante:

```
Algorithme Exo10b;  
Variables n, i, fact: Entier ;  
Début  
Répéter  
Ecrire ("Donner n≥0") ;  
Lire (n) ;  
Jusqu'à (n≥0) ;  
Si (n=0) ou (n=1)  
Alors Ecrire (n, " !=1") ;  
Sinon  
Début  
fact←n ;  
Pour i allant de n-1 à 2 pas -1  
Faire  
fact← fact*i ;  
Fait ;  
Ecrire (n, " !=", fact) ;  
Fin ;  
Fsi ;  
Fin.
```

Exercice 11:

```
Algorithme Exo11;  
Variables x, i: Entier ;  
Début  
Répéter  
Ecrire ("Donner  $1 \leq x \leq 10$ ");  
Lire (x);  
Jusqu'à ( $x \geq 1$ ) et ( $x \leq 10$ );  
Pour i allant de x à 100 pas x  
Faire  
Ecrire (i, " est un multiple de ", x);  
Fait ;  
Fin.
```

Exercice 12:

```
Algorithme Exo12;  
Variables n, i, nb1, nb2, nb3: Entier ;  
Moy : réel ;  
Début  
Répéter  
Ecrire ("Donner la valeur de  $n \leq 100$ ");  
Lire (n);  
Jusqu'à ( $n \geq 0$ ) et ( $n \leq 100$ );  
nb1  $\leftarrow$  nb2  $\leftarrow$  nb3  $\leftarrow$  0 ;  
Pour i allant de 1 à n pas 1  
Faire  
Répéter  
Ecrire ("Donner la valeur de moyenne");  
Lire (moy);  
Jusqu'à ( $moy \geq 0$ ) et ( $moy \leq 20$ );  
Si ( $moy < 10$ )  
Alors nb1  $\leftarrow$  nb1+1 ;  
Sinon Si ( $moy \geq 10$ ) et ( $moy < 15$ )  
Alors nb2  $\leftarrow$  nb2+1 ;  
Sinon nb3  $\leftarrow$  nb3+1 ;  
Fsi ;  
Fsi ;  
Fait ;  
Ecrire ("Le nombre de moyennes  $< 10$  est ", nb1) ;  
Ecrire ("Le nombre de moyennes  $\geq 10$  et  $< 15$  est ", nb2) ;  
Ecrire ("Le nombre de moyennes  $> 15$  est ", nb3) ;  
Fin.
```

Exercice 13:

Algorithme Exo13;

Variables i, nb: Entier ;
c : caractère ;

Début

nb ← 0 ;

Pour i allant de 1 à 50 pas 1

Faire

Ecrire ("Donner la valeur de caractère") ;

Lire (c) ;

Si (c='a') ou (c='A') ou (c='i') ou (c='I') ou (c='e') ou (c='E') ou (c='u') ou (c='U') ou (c='y') ou (c='Y')

Alors nb ← nb+1 ;

Fsi ;

Fait ;

Ecrire ("Le nombre de voyelles est ", nb) ;

Fin.

Exercice 14:

a) $E = \sum_{i=0}^m (2i - 1)^3$

Algorithme Exo14a;

Variables i, m, E: Entier ;

Début

Répéter

Ecrire ("Donner m ≥ 0") ;

Lire (m) ;

Jusqu'à (m ≥ 0) ;

E ← 0;

Pour i allant de 0 à m pas 1

Faire

E ← E + (2*i-1)↑3 ;

Fait ;

Ecrire ("E=", E) ;

Fin.

b) $E = 1 + \frac{1-x}{1+x} + \frac{2-x}{2+x^2} + \frac{3-x}{3+x^3} + \dots = 1 + \sum_{i \geq 1} \frac{i-x}{i+x^i}$

Algorithme Exo14b;

Variables i, m: Entier ;
E, x : Réel ;

Début

Répéter

Ecrire ("Donner m ≥ 1") ;

Lire (m) ;

Jusqu'à (m ≥ 1) ;

Répéter

Ecrire ("Donner la valeur de x") ;

Lire (x) ;

Jusqu'à (x ≥ 0) ;

E ← 1;

Pour i allant de 1 à m pas 1

Faire

E ← E + (i-x)/(i+x↑i) ;

Fait ;

Ecrire ("E=", E) ;

Fin.

$$c) E = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \dots = 1 + \sum_{i \geq 1} (-1)^i \frac{x^{2i}}{(2i)!}$$

Algorithme Exo14c;
Variables i, m, fact: Entier ;
 E, x : Réel ;
Début
Répéter
Ecrire ("Donner m ≥ 1") ;
Lire (m) ;
Jusqu'à (m ≥ 1) ;
Ecrire ("Donner la valeur de x") ;
Lire (x) ;
 E ← 1 ;
 fact ← 1 ;
Pour i allant de 1 à m pas 1
Faire
 fact ← fact * (2*i-1) * 2*i ;
 E ← E + ((-1)ⁱ * (x)^(2*i)) / fact ;
Fait ;
Ecrire ("E=", E) ;
Fin.

Exercice 15:

a) $U_1=1, U_2=2, U_n=U_{n-1}+U_{n-2}$

Algorithme Exo15a;
Variables n, i, u1, u2, u: Entier ;
Début
Répéter
Ecrire ("Donner n ≥ 3") ;
Lire (n) ;
Jusqu'à (n ≥ 3) ;
 u1 ← 1 ;
 u2 ← 2 ;
Pour i allant de 3 à n pas 1
Faire
 u ← u1 + u2 ;
 u1 ← u2 ;
 u2 ← u ;
Fait ;
Ecrire ("Le ", n, "ième terme est ", u) ;
Fin.

b) $S_0=2, S_1=3, S_2=-2, S_n=S_{n-3} + (-1)^n * S_{n-1}$

Algorithme Exo15b;
Variables n, i, S0, S1, S2, S: Entier ;
Début
Répéter
Ecrire ("Donner n ≥ 3") ;
Lire (n) ;
Jusqu'à (n ≥ 3) ;
 S0 ← 2 ;
 S1 ← 3 ;
 S2 ← -2 ;
Pour i allant de 3 à n pas 1
Faire

```

S ← S0 + (-1)i * S2 ;
S0 ← S1 ;
S1 ← S2 ;
S2 ← S ;
Fait ;
Ecrire ("Le ", n, "ième terme est ", S) ;
Fin.

```

Exercice 16:

```

Algorithme Exo16;
Variables nbval: Entier ;
           x, val : Réel;

Début
nbval ← 0 ;
Ecrire ("Donner la valeur de val") ;
Lire (val) ;
Ecrire ("Donner une valeur de x") ;
Lire (x) ;
Tant que (x ≠ 0)
Faire
Si (x > val)
Alors nbval ← nbval + 1 ;
Fsi ;
Ecrire ("Donner une valeur de x") ;
Lire (x) ;
Fait ;
Ecrire ("Le nombre de valeurs supérieures à ", val, "est ", nbval) ;
Fin.

```

Exercice 17:

```

Algorithme Exo17;
Variables nb: Entier ;
           S, x, moy : Réel;

Début
nb ← 0 ;
S ← 0 ;
Ecrire ("Donner une valeur de x") ;
Lire (x) ;
Tant que (x < 0)
Faire
nb ← nb + 1 ;
S ← S + x ;
Ecrire ("Donner une valeur de x") ;
Lire (x) ;
Fait ;
Si (nb > 0)
Alors
Début
moy ← S / nb ;
Ecrire ("La moyenne des nombres positifs ou nuls est", moy) ;
Fin ;
Sinon Ecrire ("La moyenne est 0") ;
Fsi ;
Fin.

```

Exercice 18:

1- L'algorithme qui calcule les diviseurs d'un entier :

```
Algorithme Exo18a ;  
Variables a, nb : entier ;  
Début  
Répéter  
Ecrire ("Donner la valeur de a ") ;  
Lire (a) ;  
Jusqu'à (a>0) ;  
nb←2 ;  
i←2 ;  
Tant que (i ≤ a div 2)  
Faire  
Si (a mod i=0)  
Alors nb←nb+1 ;  
Fsi ;  
i←i+1 ;  
Fait ;  
Ecrire ("Les diviseur de ", a, "est ", nb) ;  
Fin.
```

2- L'algorithme qui vérifie si un entier est parfait :

```
Algorithme Exo18b ;  
Variables a, S : entier ;  
Début  
Répéter  
Ecrire ("Donner la valeur de a ") ;  
Lire (a) ;  
Jusqu'à (a>0) ;  
S←0 ;  
i←1 ;  
Tant que (i ≤ a div 2)  
Faire  
Si (a mod i=0)  
Alors S←S+i ;  
Fsi ;  
i←i+1 ;  
Fait ;  
Si (a=S)  
Alors Ecrire ("Le nombre ", a, "est parfait") ;  
Sinon Ecrire ("Le nombre ", a, "n'est pas parfait") ;  
Fsi ;  
Fin.
```

3- L'algorithme qui vérifie si un entier est premier :

```
Algorithme Exo18c ;  
Variables a, nb : entier ;  
Début  
Répéter  
Ecrire ("Donner la valeur de a ") ;  
Lire (a) ;  
Jusqu'à (a>0) ;  
nb←2;  
i←2 ;  
Tant que (i ≤ a div 2)  
Faire  
Si (a mod i=0)  
Alors nb←nb+1 ;  
Fsi ;  
i←i+1 ;  
Fait ;  
Si (nb=2)  
Alors Ecrire ("Le nombre ", a, "est premier") ;  
Sinon Ecrire ("Le nombre ", a, "n'est pas premier") ;  
Fsi ;  
Fin.
```

Exercice 19:

1- L'algorithme qui calcule A*B sans utiliser l'opérateur de multiplication * :

```
Algorithme Exo19a ;  
Variables A, B, P, i : entier ;  
Début  
Répéter  
Ecrire ("donner deux entiers naturels positifs");  
Lire (A, B);  
Jusqu'à (A≥0) et (B≥0) ;  
Si (A=0) ou (B=0)  
Alors Ecrire ("Le produit est nul") ;  
Sinon  
Début  
P←0 ;  
Pour i allant de 1 à B pas 1  
Faire  
P←P+A ;  
Fait ;  
Ecrire ("Le produit entre ",A," et ",B," = ",P) ;  
Fin ;  
Fsi ;  
Fin.
```

2- L'algorithme qui calcule A/B sans utiliser l'opérateur de division / :

```
Algorithme Exo19b ;  
Variables A, B, Q, R, i : entier ;  
Début  
Répéter  
Ecrire ("donner deux entiers naturels positifs");  
Lire (A, B);  
Jusqu'à (A≥0) et (B>0) ;  
Q←0 ;  
R←A ;  
Tant que (R≥B)  
Faire  
Q←Q+1 ;  
R←R-B ;  
Fait ;  
Ecrire ("Le quotient Q= ",Q, "Le reste R= ", R) ;  
Fin ;  
Fin.
```

Exercice 20:

1- L'algorithme qui convertit un nombre décimal A en binaire :

```
Algorithme Exo20a ;  
Variables A, B, i : entier ;  
ch : chaîne de caractères ;  
Début  
Répéter  
Ecrire ("donner un entier positif A");  
Lire (A);  
Jusqu'à (A≥0) ;  
B←A ;  
ch←"" ;  
Faire  
Si (B mod 2=0)  
Alors ch← "0"&ch ;  
Sinon ch← "1"&ch ;  
Fsi ;  
B←B div 2 ;  
Tant que (B≠0);  
Ecrire ("La conversion de ", A, "est ", ch) ;  
Fin ;  
Fin.
```

2- L'algorithme qui vérifie si un entier A est dans une base B :

```
Algorithme Exo19b ;  
Variables A, B, C, i : entier ;  
          Bol : booléen;  
Début  
Répéter  
Ecrire ("donner une base B ");  
Lire (B);  
Jusqu'à (B≥1) et (B≤11) ;  
Répéter  
Ecrire ("donner un entier A positif ou nul ");  
Lire (A);  
Jusqu'à (A≥0) ;  
Bol ← Faux ;  
C ← A ;  
Faire  
Si (C mod 10 > B)  
Alors Bol ← Vrai ;  
C ← C div 10 ;  
Tant que (B=Faux) et (C≠0) ;  
Si (Bol=Faux)  
Alors Ecrire ("Le nombre ", A, " appartient à la base ", B) ;  
Sinon Ecrire ("Le nombre ", A, " n'appartient pas à la base ", B) ;  
Fsi ;  
Fin ;  
Fin.
```

Série d'exercices n° 5 (Tableaux)

Exercice 1: Ecrire un algorithme pour déterminer la plus petite valeur d'un tableau T d'entiers de dimension n ($n \leq 20$) ainsi que sa position.

Solution :

➤ Algorithme :

Algorithme exercice1 ;
Variation T : tableau[20] entier ;
 n, i, posmin, min : entier ;
Début
 {Lecture de la taille et vérification n ($n > 0$ et $n \leq 20$)}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à ($n > 0$) et ($n \leq 20$) ;
 {Lecture des éléments de tableau}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément d'indice i=", i) ;
Lire(T[i]) ;
Fait ;
 min ← T[1] ; posmin ← 1 ; {Supposer que le premier élément est le minimum}
 {Vérifier si l'élément supposé comme min l'est toujours}
Pour i allant de 2 à n pas 1
Faire
Si (T[i] < min) **alors début** min ← T[i] ; posmin ← i ; **fin** ; **fsi** ; {si on trouve un élément plus petit que ce min on mis à jour le min}
Fait ;
 {Affichage de l'élément minimum}
Ecrire ("Le minimum de ce tableau est ", "min, de position égale à ", posmin) ;
Fin.

➤ Déroulement :

Actions	n	i	T[1]	T[2]	T[3]	T[4]	min	posmin
Lire(n) ;	4	?	?	?	?	?	?	?
i ← 1 ;	4	1	?	?	?	?	?	?
Lire(T[i]) ;	4	1	0	?	?	?	?	?
i ← i+1 ;	4	2	0	?	?	?	?	?
Lire(T[i]) ;	4	2	0	3	?	?	?	?
i ← i+1 ;	4	3	0	3	?	?	?	?
Lire(T[i]) ;	4	3	0	3	-1	?	?	?
i ← i+1 ;	4	4	0	3	-1	?	?	?
Lire(T[i]) ;	4	4	0	3	-1	2	?	?
i ← i+1 ;	4	5	0	3	-1	2	?	?
min ← T[1] ;	4	5	0	3	-1	2	0	?
posmin ← i ;	4	5	0	3	-1	2	0	1
i ← 2 ;	4	2	0	3	-1	2	0	1
i ← i+1 ;	4	3	0	3	-1	2	0	1
min ← T[i] ;	4	3	0	3	-1	2	-1	1
posmin ← i ;	4	3	0	3	-1	2	-1	3
i ← i+1 ;	4	4	0	3	-1	2	-1	3
i ← i+1 ;	4	5	0	3	-1	2	-1	3

Le minimum de ce tableau est -1 de position égale à 3.

Exercice 2: Ecrire un algorithme qui calcule le plus grand écart dans un tableau V de taille n ($n \leq 10$) d'entiers (l'écart est la valeur absolue de la différence de deux éléments).

Solution :

➤ Algorithme :

```

Algorithme exercice2 ;
Variables T : tableau[10] entier ;
           n,i, min, max, ecart : entier ;

Début
{Lecture de la taille et vérification n ( $n > 0$  et  $n \leq 10$ )}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à ( $n > 0$ ) et ( $n \leq 10$ ) ;
{Lecture des éléments de tableau}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément d'indice i=",i) ;
Lire(T[i]) ;
Fait ;
min ← T[1] ; max ← T[1] ; {Supposer que le premier élément est le minimum et le maximum}
{Vérifier si l'élément supposé comme min et max sont toujours}
Pour i allant de 2 à n pas 1
Faire
Si (T[i] < min) alors min ← T[i] ; fsi ; {si on trouve un élément plus petit que ce min on mit à jour le min}
Si (T[i] > max) alors max ← T[i] ; fsi ; {si on trouve un élément plus grand que ce max on mit à jour le max}
Fait ;
{Le plus grand écart est la valeur absolue de différence entre le max et le min}
ecart ← max-min;
Si (ecart < 0) alors ecart ← (-1)*ecart; fsi ;
{Affichage de l'écart}
Ecrire ("L'écart de ce tableau est ", ecart) ;
Fin.

```

➤ Déroulement :

Actions	n	i	T[1]	T[2]	T[3]	T[4]	min	max	ecart
Lire(n) ;	4	?	?	?	?	?	?	?	?
i ← 1 ;	4	1	?	?	?	?	?	?	?
Lire(T[i]) ;	4	1	0	?	?	?	?	?	?
i ← i+1 ;	4	2	0	?	?	?	?	?	?
Lire(T[i]) ;	4	2	0	3	?	?	?	?	?
i ← i+1 ;	4	3	0	3	?	?	?	?	?
Lire(T[i]) ;	4	3	0	3	-1	?	?	?	?
i ← i+1 ;	4	4	0	3	-1	?	?	?	?
Lire(T[i]) ;	4	4	0	3	-1	2	?	?	?
i ← i+1 ;	4	5	0	3	-1	2	?	?	?
min ← T[1] ;	4	5	0	3	-1	2	0	?	?
max ← T[1] ;	4	5	0	3	-1	2	0	0	?
i ← 1 ;	4	1	0	3	-1	2	0	0	?
max ← T[i] ;	4	1	0	3	-1	2	0	3	?
i ← i+1 ;	4	2	0	3	-1	2	0	3	?
i ← i+1 ;	4	3	0	3	-1	2	0	3	?
min ← T[i] ;	4	3	0	3	-1	2	-1	3	?
i ← i+1 ;	4	4	0	3	-1	2	-1	3	?
i ← i+1 ;	4	5	0	3	-1	2	-1	3	?
ecart ← max-min ;	4	5	0	3	-1	2	-1	3	4

L'écart de ce tableau est 4.

Exercice 3: Écrire un algorithme qui remplit un tableau par 100 entiers lus au clavier. Ajoutez 1 à toutes les valeurs de rang pair de ce tableau. Soustrayez 1 à toutes les valeurs de rang impair. Affichez l'intégralité de ce tableau.

Solution :

➤ Algorithme :

```

Algorithme exercice3 ;
Variables T : tableau[100] entier ;
           i : entier ;

Début
  {Lecture des éléments de tableau}
  Pour i allant de 1 à 100 pas 1
  Faire
  Ecrire ("Donner la valeur de l'élément d'indice i=",i) ;
  Lire(T[i]) ;
  Fait ;
  {Modification de Tableau}
  Pour i allant de 1 à 100 pas 1
  Faire
  Si (T[i] mod 2=0) alors T[i]←T[i]+1 ; {mod calcule le reste de la division (si le reste de division est 0 donc il est pair)}
           sinon T[i]←T[i]-1 ; { T[i] est impair}
  fsi ;
  Fait ;
  {Affichage des éléments de tableau après modification}
  Pour i allant de 1 à 100 pas 1
  Faire
  Ecrire ("L'élément d'indice i=", i, "est ", T[i]) ;
  Fait ;
Fin.

```

➤ Déroulement : pour un tableau de 4 éléments

Actions	i	T[1]	T[2]	T[3]	T[4]
Lire(n) ;	?	?	?	?	?
i←1 ;	1	?	?	?	?
Lire(T[i]) ;	1	0	?	?	?
i←i+1 ;	2	0	?	?	?
Lire(T[i]) ;	2	0	3	?	?
i←i+1 ;	3	0	3	?	?
Lire(T[i]) ;	3	0	3	-1	?
i←i+1 ;	4	0	3	-1	?
Lire(T[i]) ;	4	0	3	-1	2
i←1 ;	1	0	3	-1	2
T[i]← T[i]+1 ;	1	1	3	-1	2
i←i+1 ;	2	1	3	-1	2
T[i]← T[i]-1 ;	2	1	2	-1	2
i←i+1 ;	3	1	2	-1	2
T[i]← T[i]-1 ;	3	1	2	-2	2
i←i+1 ;	4	1	2	-2	2
T[i]← T[i]+1 ;	4	1	2	-2	1
i←i+1 ;	5	1	2	-2	1

➤ Affichage de Tableau :

L'élément d'indice i=1 est 1

L'élément d'indice i=2 est 2

L'élément d'indice i=3 est -2

L'élément d'indice i=4 est 1

Exercice 4: Ecrire un algorithme qui calcul le produit scalaire de deux vecteurs d'entiers U et V de même dimension n (n<=10) lus au clavier.

Solution :

➤ Algorithmes :

```

Algorithme exercice4 ;
Variables V, U : tableau[10] Réel ;
           i, n: entier ;
           PS : Réel ;

Début
{Lecture de la taille et vérification n (n>0 et n≤10)}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à (n>0) et (n≤10) ;
{Lecture des éléments de tableau V}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau V d'indice i=", i) ;
Lire(V[i]) ;
Fait ;
{Lecture des éléments de tableau U}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau U d'indice i=", i) ;
Lire(U[i]) ;
Fait ;
{Calcul de Produit Scalaire}
PS←0 ;
Pour i allant de 1 à n pas 1
Faire
PS←PS+V[i]*U[i] ;
Fait ;
{Affichage de produit scalaire}
Ecrire ("Le produit scalaire est ", PS) ;
Fin.

```

➤ Déroulement :

Actions	N	i	V[1]	V[2]	V[3]	V[4]	U[1]	U[2]	U[3]	U[4]	PS
Lire(n) ;	4	?	?	?	?	?	?	?	?	?	?
i←1 ;	4	1	?	?	?	?	?	?	?	?	?
Lire(V[i]) ;	4	1	0	?	?	?	?	?	?	?	?
i←i+1 ;	4	2	0	?	?	?	?	?	?	?	?
Lire(V[i]) ;	4	2	0	3	?	?	?	?	?	?	?
i←i+1 ;	4	3	0	3	?	?	?	?	?	?	?
Lire(V[i]) ;	4	3	0	3	-1	?	?	?	?	?	?
i←i+1 ;	4	4	0	3	-1	?	?	?	?	?	?
Lire(V[i]) ;	4	4	0	3	-1	2	?	?	?	?	?
i←i+1 ;	4	5	0	3	-1	2	?	?	?	?	?
i←1 ;	4	1	0	3	-1	2	?	?	?	?	?
Lire(U[i]) ;	4	1	0	3	-1	2	1	?	?	?	?
i←i+1 ;	4	2	0	3	-1	2	1	?	?	?	?
Lire(U[i]) ;	4	2	0	3	-1	2	1	2	?	?	?
i←i+1 ;	4	3	0	3	-1	2	1	2	?	?	?
Lire(U[i]) ;	4	3	0	3	-1	2	1	2	5	?	?
i←i+1 ;	4	4	0	3	-1	2	1	2	5	?	?
Lire(U[i]) ;	4	4	0	3	-1	2	1	2	5	0	?
i←i+1 ;	4	5	0	3	-1	2	1	2	5	0	?
PS←0 ;	4	5	0	3	-1	2	1	2	5	0	0
i←1 ;	4	1	0	3	-1	2	1	2	5	0	0

PS←PS+V[i]*U[i];	4	1	0	3	-1	2	1	2	5	0	0
i←i+1;	4	2	0	3	-1	2	1	2	5	0	0
PS←PS+V[i]*U[i];	4	2	0	3	-1	2	1	2	5	0	6
i←i+1;	4	3	0	3	-1	2	1	2	5	0	6
PS←PS+V[i]*U[i];	4	3	0	3	-1	2	1	2	5	0	1
i←i+1;	4	4	0	3	-1	2	1	2	5	0	1
PS←PS+V[i]*U[i];	4	4	0	3	-1	2	1	2	5	0	1
i←i+1;	4	5	0	3	-1	2	1	2	5	0	1

➤ **Affichage :** Le produit scalaire est **1**

Exercice 5: soit un tableau T1 de dimension n ($n \leq 50$) et contenant que des 0 et des 1, écrire un algorithme qui permet de construire le tableau T2 représentant le complément à 1 de T1. (Exemple : Le complément à 1 de 11000111 → est 00111000).

Solution :

➤ **Algorithme :**

```

Algorithme exercice5 ;
Variables T1, T2 : tableau[50] entier ;
           i, n: entier ;

Début
  {Lecture de la taille et vérification n (n>0 et n≤50)}
  Répéter
  Ecrire ("Donner la valeur de taille n") ;
  Lire(n) ;
  Jusqu'à (n>0) et (n≤50) ;
  {Lecture des éléments de tableau T1}
  Pour i allant de 1 à n pas 1
  Faire
  Répéter
  Ecrire ("Donner la valeur de l'élément de tableau T1 d'indice i=", i, " de valeur 1 ou 0 ") ;
  Lire(T1[i]) ;
  jusqu'à (T1[i]=0) ou (T1[i]=1) ;
  Fait ;
  {Calcul de T2 le complément T1}
  Pour i allant de 1 à n pas 1
  Faire
  Si (T1[i]=1) alors T2[i]←0 ;
           sinon T2[i]←1 ;
  fsi ;
  Fait ;
  {Affichage de Tableau T1}
  Pour i allant de 1 à n pas 1
  Faire
  Ecrire ("L'élément d'indice i=", i, " de tableau T1 est ", T1[i]) ;
  Fait ;
  {Affichage de Tableau T2}
  Pour i allant de 1 à n pas 1
  Faire
  Ecrire ("L'élément d'indice i=", i, " de tableau T2 est ", T2[i]) ;
  Fait ;
Fin.

```

➤ Déroulement :

Actions	N	i	T1[1]	T1[2]	T1[3]	T1[4]	T2[1]	T2[2]	T2[3]	T2[4]
Lire(n) ;	4	?	?	?	?	?	?	?	?	?
i←1 ;	4	1	?	?	?	?	?	?	?	?
Lire(T1[i]) ;	4	1	1	?	?	?	?	?	?	?
i←i+1 ;	4	2	0	?	?	?	?	?	?	?
Lire(T1[i]) ;	4	2	1	1	?	?	?	?	?	?
i←i+1 ;	4	3	1	1	?	?	?	?	?	?
Lire(T1[i]) ;	4	3	1	1	0	?	?	?	?	?
i←i+1 ;	4	4	1	1	0	?	?	?	?	?
Lire(T1[i]) ;	4	4	1	1	0	0	?	?	?	?
i←i+1 ;	4	5	1	1	0	0	?	?	?	?
i←1 ;	4	1	1	1	0	0	?	?	?	?
T2[i]←0 ;	4	1	1	1	0	0	0	?	?	?
i←i+1 ;	4	2	1	1	0	0	0	?	?	?
T2[i]←0 ;	4	2	1	1	0	0	0	0	?	?
i←i+1 ;	4	3	1	1	0	0	0	0	?	?
T2[i]←1 ;	4	3	1	1	0	0	0	0	1	?
i←i+1 ;	4	4	1	1	0	0	0	0	1	?
T2[i]←1 ;	4	4	1	1	0	0	0	0	1	1
i←i+1 ;	4	5	1	1	0	0	0	0	1	1

➤ Affichage :

L'élément d'indice i=1 de tableau T1 est 1
 L'élément d'indice i=2 de tableau T1 est 1
 L'élément d'indice i=3 de tableau T1 est 0
 L'élément d'indice i=4 de tableau T1 est 0
 L'élément d'indice i=1 de tableau T2 est 0
 L'élément d'indice i=2 de tableau T2 est 0
 L'élément d'indice i=3 de tableau T2 est 1
 L'élément d'indice i=4 de tableau T2 est 1

Exercice 6: Soit un vecteur X ($N \leq 20$), écrire un algorithme qui calcule VMEP (valeur moyenne des éléments positifs), VMEN (Valeur Moyenne des éléments négatifs) et NEN (Nombre d'éléments nuls).

Solution :

➤ Algorithme :

```

Algorithme exercice- ;
Variables X : tableau[20] Entier ;
           i, n, NEN, NBVP, NBVN, SVP, SVN : entier ;
           VMEP, VMEN: Réel ;

Début
{Lecture de la taille et vérification n (n>0 et n≤20)}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à (n>0) et (n≤20) ;
{Lecture des éléments de tableau X}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau X d'indice i=", i) ;
Lire(X[i]) ;
Fait ;
{Calcul de VMEP, VMEN, NEN, NBVP, NBVN, SVP, SVN}
NEN←0 ; {initialiser le nombre de valeurs nuls à 0}
NBVP← 0; {initialiser le nombre de valeurs positives à 0}
NBVN←0 ; {initialiser le nombre de valeurs négatives à 0}
SVP← 0; {initialiser le nombre la somme des valeurs positives à 0}
SVN← 0; {initialiser le nombre la somme des valeurs négatives à 0}
  
```

Pour i allant de 1 à n pas 1

Faire

Si (X[i]=0) alors NEN←NEN+1 ;
 sinon si (X[i]>0) alors

début
 NBVP ← NBVP +1 ;
 SVP ← SVP +X[i] ;
 fin ;

sinon

début
 NBVN ← NBVN +1 ;
 SVN ← SVN +X[i] ;
 fin ;

fsi ;

fsi

Fait ;

VMEP←SVP/NBVP ; {La moyenne des éléments positifs est la division de la somme des éléments positifs sur le nombre des éléments positifs}

VMEN←SVN/NBVN ; {La moyenne des éléments négatifs est la division de la somme des éléments négatifs sur le nombre des éléments négatifs}

{Affichage des résultats}

Ecrire ("La valeur moyenne des éléments positifs est : ", VMEP) ;

Ecrire ("La valeur moyenne des éléments négatifs est : ", VMEN) ;

Ecrire ("Le nombre des éléments nuls est : ", NEN) ;

Fin.

➤ Déroulement :

Actions	N	i	X[1]	X[2]	X[3]	X[4]	NEN	NBVP	SVP	NBVN	SVN	VMEP	VMEN
Lire(n) ;	4	?	?	?	?	?	?	?	?	?	?	?	?
i←1 ;	4	1	?	?	?	?	?	?	?	?	?	?	?
Lire(X[i]) ;	4	1	0	?	?	?	?	?	?	?	?	?	?
i←i+1 ;	4	2	0	?	?	?	?	?	?	?	?	?	?
Lire(X[i]) ;	4	2	0	3	?	?	?	?	?	?	?	?	?
i←i+1 ;	4	3	0	3	?	?	?	?	?	?	?	?	?
Lire(X[i]) ;	4	3	0	3	-1	?	?	?	?	?	?	?	?
i←i+1 ;	4	4	0	3	-1	?	?	?	?	?	?	?	?
Lire(X[i]) ;	4	4	0	3	-1	2	?	?	?	?	?	?	?
i←i+1 ;	4	5	0	3	-1	2	?	?	?	?	?	?	?
NEN←0 ;	4	5	0	3	-1	2	0	?	?	?	?	?	?
NBVP←0 ;	4	5	0	3	-1	2	0	0	?	?	?	?	?
NBVN←0 ;	4	5	0	3	-1	2	0	0	?	0	?	?	?
SVP←0 ;	4	5	0	3	-1	2	0	0	0	?	?	?	?
SVN←0 ;	4	5	0	3	-1	2	0	0	0	0	0	?	?
i←1 ;	4	1	0	3	-1	2	0	0	0	0	0	?	?
NEN←NEN+1 ;	4	1	0	3	-1	2	1	0	0	0	0	?	?
i←i+1 ;	4	2	0	3	-1	2	1	0	0	0	0	?	?
NBVP←NBVP+1 ;	4	2	0	3	-1	2	1	1	0	0	0	?	?
SVP←SVP+X[i] ;	4	2	0	3	-1	2	1	1	3	0	0	?	?
i←i+1 ;	4	3	0	3	-1	2	1	1	3	0	0	?	?
NBVN←NBVN+1 ;	4	3	0	3	-1	2	1	1	3	1	0	?	?
SVN←SVN+X[i] ;	4	3	0	3	-1	2	1	1	3	1	-1	?	?
i←i+1 ;	4	4	0	3	-1	2	1	1	3	1	-1	?	?
NBVP←NBVP+1 ;	4	4	0	3	-1	2	1	2	3	1	-1	?	?
SVP←SVP+X[i] ;	4	4	0	3	-1	2	1	2	5	1	-1	?	?
i←i+1 ;	4	5	0	3	-1	2	1	2	5	1	-1	?	?
VMEP←SVP/NBVP ;	4	5	0	3	-1	2	1	2	5	1	-1	2.5	?
VMEN←SVN/NBVN ;	4	5	0	3	-1	2	1	2	5	1	-1	2.5	-1

➤ Affichage :

La valeur moyenne des éléments positifs est : **2.5**

La valeur moyenne des éléments négatifs est : **-1**

Le nombre des éléments nuls est : **1**

Exercice 7: Ecrire un algorithme qui renverse un tableau T de taille n ($n \leq 50$) remplis au clavier par des éléments réels (la 1ère valeur devient la dernière, la 2ème valeur devient avant dernière).

Solution :

➤ Algorithme :

```

Algorithme exercice7 ;
Variables T : tableau[50] entier ;
           i, n, x: entier ;

Début
{Lecture de la taille et vérification n ( $n > 0$  et  $n \leq 50$ )}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à ( $n > 0$ ) et ( $n \leq 50$ ) ;
{Lecture des éléments de tableau T}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau T d'indice i=", i) ;
Lire(T[i]) ;
Fait ;
{Calcul de l'inverse de T}
i ← 1 ;
Tant que ( $i \leq n \text{ div } 2$ )
Faire
X ← T[i] ;
T[i] ← T[n-i+1] ;
T[n-i+1] ← X ;
i ← i+1 ;
Fait ;
{Affichage de Tableau T après inversement}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("L'élément d'indice i=", i, " de tableau T est ", T[i]) ;
Fait ;
Fin.

```

➤ Déroulement :

Actions	N	i	T[1]	T[2]	T[3]	T[4]	X
Lire(n) ;	4	?	?	?	?	?	?
i ← 1 ;	4	1	?	?	?	?	?
Lire(T[i]) ;	4	1	0	?	?	?	?
i ← i+1 ;	4	2	0	?	?	?	?
Lire(T[i]) ;	4	2	0	3	?	?	?
i ← i+1 ;	4	3	0	3	?	?	?
Lire(T[i]) ;	4	3	0	3	-1	?	?
i ← i+1 ;	4	4	0	3	-1	?	?
Lire(T[i]) ;	4	4	0	3	-1	2	?
i ← i+1 ;	4	5	0	3	-1	2	?
i ← 1 ;	4	1	0	3	-1	2	?
X ← T[i] ;	4	1	0	3	-1	2	0
T[i] ← T[n-i+1] ;	4	1	2	3	-1	2	0
T[n-i+1] ← X ;	4	1	2	3	-1	0	0
i ← i+1 ;	4	2	2	3	-1	0	0
X ← T[i] ;	4	2	2	-1	0	0	3
T[i] ← T[n-i+1] ;	4	2	2	-1	0	0	3
T[n-i+1] ← X ;	4	2	2	-1	3	0	3
i ← i+1 ;	4	3	2	-1	3	0	3

➤ Affichage :

L'élément d'indice i=1 de tableau T est 2

L'élément d'indice i= 2 de tableau T est -1

L'élément d'indice i=3 de tableau T est 3

L'élément d'indice i=4 de tableau T est 0

Exercice 8: Soit un vecteur X ($N \leq 20$), écrire un algorithme qui définit les éléments du vecteur Y ($N \leq 20$)

et les affiche en sachant que : $y_i = \frac{1}{N} \sum_{j=1}^i x_j$ (**Devoir**)

Solution :

➤ Algorithme :

Algorithme exercice8 ;

Variables X, Y : tableau[20] Réel ;

i, n: entier ;

Début

{Lecture de la taille et vérification n ($n > 0$ et $n \leq 20$)}

Répéter

Ecrire ("Donner la valeur de taille n") ;

Lire(n) ;

Jusqu'à ($n > 0$) et ($n \leq 20$) ;

{Lecture des éléments de tableau X}

Pour i allant de 1 à n pas 1

Faire

Ecrire ("Donner la valeur de l'élément de tableau TX d'indice i=", i) ;

Lire(X[i]) ;

Fait ;

$Y[1] \leftarrow X[1]/n$;

{Affichage de s occurrences de val et leur position}

Pour i allant de 1 à n pas 1

Faire

$Y[i] \leftarrow Y[i-1] + X[i]/n$

Fait ;

{Affichage de tableau X}

Pour i allant de 1 à n pas 1

Faire

Ecrire ("L'élément d'indice i=", i, " de tableau X est ", X[i]) ;

Fait ;

{Affichage de tableau Y}

Pour i allant de 1 à n pas 1

Faire

Ecrire ("L'élément d'indice i=", i, " de tableau Y est ", Y[i]) ;

Fait ;

Fin.

Exercice 9: Rechercher un élément dans un tableau T de réels et de taille n ($n \leq 50$) séquentiellement connaissant sa valeur. Si la valeur existe, afficher toutes les occurrences et leurs positions sinon afficher un message cette valeur n'existe pas. **(Devoir)**

Solution :

➤ **Algorithme :**

```
Algorithme exercice9 ;
Variables T : tableau[50] Réel ;
          i, n: entier ;
          val : Réel ;
          Bol : booléen ;

Début
{Lecture de la taille et vérification n ( $n > 0$  et  $n \leq 50$ )}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à ( $n > 0$ ) et ( $n \leq 50$ ) ;
Ecrire ("Donner la valeur de val") ;
Lire(val) ;
{Lecture des éléments de tableau T}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau T d'indice i=", i) ;
Lire(T[i]) ;
Fait ;
Bol ← Faux ;
{Affichage de s occurrences de val et leur position}
Pour i allant de 1 à n pas 1
Faire
Si (T (i)=val)
Alors
Début
Ecrire ("L'élément d'indice i=", i, " de tableau T est ", T[i], " est une occurrence de ", val) ;
Bol ← Vrai ;
Fin ;
Fsi ;
Fait ;
Si (Bol =Faux)
Alors Ecrire ("La valeur val=", val, " n'existe pas dans le tableau T") ;
Fsi ;
Fin.
```

Solution :

➤ Algorithme :

```
Algorithme exercice9 ;
Variables T : tableau[50] Réel ;
           i, n: entier ;
           val : Réel ;
           Bol : booléen ;

Début
{Lecture de la taille et vérification n (n>0 et n≤50)}
Répéter
Ecrire ("Donner la valeur de taille n") ;
Lire(n) ;
Jusqu'à (n>0) et (n≤50) ;
Ecrire ("Donner la valeur de val") ;
Lire(val) ;
{Lecture des éléments de tableau T}
Pour i allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de tableau T d'indice i=", i) ;
Lire(T[i]) ;
Fait ;
Bol ←Faux ;
{Affichage de s occurrences de val et leur position}
Pour i allant de 1 à n pas 1
Faire
Si (T (i)=val)
Alors
Début
Ecrire ("L'élément d'indice i=", i, " de tableau T est ", T[i], " est une occurrence de ", val) ;
Bol←Vrai ;
Fin ;
Fsi ;
Fait ;
Si (Bol =Faux)
Alors Ecrire ("La valeur val=", val, " n'existe pas dans le tableau T") ;
Fsi ;
Fin.
```

Exercice 10: Ecrire un algorithme pour une matrice A d'entier qui :

- Lit les dimensions **n** et **m** du matrice A et vérifie qu'elles sont inférieures à 10 ; sinon il affichera :
« Le nombre de lignes ou de colonnes est incorrect » et sortira du l'algorithme.
- Remplit la matrice par des valeurs entrées au clavier et affiche le tableau.
- Affiche la somme, le produit, la moyenne de tous les éléments, ainsi que le maximum et le minimum.
- Affiche la somme de chaque ligne.
- Transfère la matrice dans un tableau V à une dimension n*m et afficher V.

Solution :

- Algorithme pour la vérification des indices ainsi que la lecture et l'affichage de la matrice A :

```
Algorithme exercice10a ;
Variables A : tableau[10,10] entier ;
           i, j, n, m: entier ;
Début
{Lecture de nombre de lignes ainsi que le nombre de colonnes de la matrice A}
Ecrire ("Donner le nombre de lignes n≤10 et nombre de colonnes m≤10") ;
Lire(n) ;
Si (n<0) ou (n>10) ou (m<0) ou (m>10) alors écrire ("nombre de colonnes ou lignes incorrect")
      Sinon
        Début
          {Lecture des éléments de la matrice A}
          Pour i allant de 1 à n pas 1
            Faire
              Pour j allant de 1 à m pas 1
                Faire
                  Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
                  Lire(A[i][j]) ;
                  Fait ;
                Fait ;
              {Affichage des éléments de la matrice A}
              Pour i allant de 1 à n pas 1
                Faire
                  Pour j allant de 1 à m pas 1
                    Faire
                      Ecrire ("L'élément de ligne i=", i, " et colonne j=", j, "est", A[i][j]) ;
                      Fait ;
                    Fait ;
                  Fin ;
                Fait ;
              Fin ;
            Fait ;
          Fin ;
        Fsi ;
      Fin.
```

➤ Algorithme pour le calcul de la somme, le produit, la moyenne de tous les éléments:

```
Algorithme exercice10b ;
Variables A : tableau[10,10] entier ;
           i, j, n, m, som, prod: entier ;
           moy: réel;

Début
{Lecture de nombre de lignes ainsi que le nombre de colonnes de la matrice A}
Ecrire ("Donner le nombre de lignes n≤10 et nombre de colonnes m≤10") ;
Lire(n) ;
Si (n<0) ou (n>10) ou (m<0) ou (m>10) alors écrire ("nombre de colonnes ou lignes incorrect")
Sinon
Début
{Lecture des éléments de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à m pas 1
Faire
Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
Lire(A[i][j]) ;
Fait ;
Fait ;
som←0 ; {initialiser la somme à la valeur 0}
prod←1 ; {initialiser le produit à la valeur 1}
{Calcul de la somme et le produit}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à m pas 1
Faire
som←som+A[i][j] ; {ajouter à la somme la valeur de l'élément A[i][j]}
prod←prod*A[i][j] ; {Multiplier le produit par la valeur de l'élément A[i][j]}
Fait ;
Fait ;
{Calcul de la moyenne}
moy←som/(n*m) ;
{Affichage des résultats}
Ecrire ("La somme de tous les éléments", som) ;
Ecrire ("Le produit de tous les éléments", prod) ;
Ecrire ("La moyenne de tous les éléments", moy) ;
Fin ;

Fsi ;
Fin.
```

➤ Algorithme pour le calcul de maximum et le minimum de la matrice A :

Algorithme exercice10c ;

Variables A : tableau[10,10] **entier** ;

i, j, n, m, max, min: **entier** ;

Début

{Lecture de nombre de lignes ainsi que le nombre de colonnes de la matrice A}

Ecrire ("Donner le nombre de lignes $n \leq 10$ et nombre de colonnes $m \leq 10$ ") ;

Lire(n) ;

Si ($n < 0$) ou ($n > 10$) ou ($m < 0$) ou ($m > 10$) **alors écrire** ("nombre de colonnes ou lignes incorrect")

Sinon

Début

{Lecture des éléments de la matrice A}

Pour i allant de 1 à n pas 1

Faire

Pour j allant de 1 à m pas 1

Faire

Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;

Lire(A[i][j]) ;

Fait ;

Fait ;

$\max \leftarrow A[1][1]$; {initialiser le maximum par le premier élément}

$\min \leftarrow A[1][1]$; {initialiser le minimum par le premier élément}

Pour i allant de 1 à n pas 1

Faire

Pour j allant de 1 à m pas 1

Faire

Si ($A[i][j] > \max$) **alors** $\max \leftarrow A[i][j]$; **fsi** ;

Si ($A[i][j] < \min$) **alors** $\min \leftarrow A[i][j]$; **fsi** ;

Fait ;

Fait ;

{Affichage des résultats}

Ecrire ("Le maximum de A est ", max) ;

Ecrire ("Le minimum de A est ", min) ;

Fin ;

Fsi ;

Fin.

➤ Algorithme pour le calcul de la somme de chaque ligne :

```
Algorithme exercice10d ;
Variables A : tableau[10,10] entier ;
           i, j, n, m, soml: entier ;
           Début
{Lecture de nombre de lignes ainsi que le nombre de colonnes de la matrice A}
Ecrire ("Donner le nombre de lignes  $n \leq 10$  et nombre de colonnes  $m \leq 10$ ") ;
Lire(n) ;
Si (n<0) ou (n>10) ou (m<0) ou (m>10) alors écrire ("nombre de colonnes ou lignes incorrect")
           Sinon
           Début
           {Lecture des éléments de la matrice A}
           Pour i allant de 1 à n pas 1
           Faire
           Pour j allant de 1 à m pas 1
           Faire
           Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
           Lire(A[i][j]) ;
           Fait ;
           Fait ;
           Pour i allant de 1 à n pas 1
           Faire
           soml←0; {initialiser la somme à 0}
           Pour j allant de 1 à m pas 1
           Faire
           soml←soml+ A[i][j]; {Ajouter l'élément A[i][j] à soml }
           Fait ;
           Ecrire ("écrire la somme de la ligne i= ", i, "est", soml) ;
           Fait ;
           Fin ;
Fsi ;
Fin.
```

➤ Algorithme qui transfère la matrice dans un tableau V à une dimension n*m et afficher V:

```
Algorithme exercice10d ;  
Variables A : tableau[10,10] entier ;  
          i, j, k, n, m: entier ;  
          V: tableau[100] entier;  
  Début  
  {Lecture de nombre de lignes ainsi que le nombre de colonnes de la matrice A}  
  Ecrire ("Donner le nombre de lignes n≤10 et nombre de colonnes m≤10") ;  
  Lire(n) ;  
  Si (n<0) ou (n>10) ou (m<0) ou (m>10) alors écrire ("nombre de colonnes ou lignes incorrect")  
    Sinon  
    Début  
    {Lecture des éléments de la matrice A}  
    Pour i allant de 1 à n pas 1  
    Faire  
    Pour j allant de 1 à m pas 1  
    Faire  
    Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;  
    Lire(A[i][j]) ;  
    Fait ;  
    Fait ;  
    k←1 ;  
    Pour i allant de 1 à n pas 1  
    Faire  
    Pour j allant de 1 à m pas 1  
    Faire  
    V[k]← A[i][j];  
    k←k+1 ;  
    Fait ;  
    Fait ;  
    {Affichage des éléments de tableau V}  
    Pour i allant de 1 à k pas 1  
    Faire  
    Ecrire ("L'élément d'indice i=", i, " est ", V[i]) ;  
    Fait ;  
    Fin ;  
Fsi ;  
Fin.
```

Exercice 11: Ecrire un algorithme qui calcule la somme des éléments de la diagonale principale d'une matrice carrée A d'entier et de dimension n ($n \leq 20$). Les éléments de la matrice A ainsi que la dimension seront lus au clavier.

Solution :

➤ Algorithme :

```
Algorithme exercice11 ;
Variables A : tableau[20,20] entier ;
          n,i, som : entier ;
Début
{Lecture de la dimension et vérification n ( $n > 0$  et  $n \leq 20$ )}
Répéter
Ecrire ("Donner la valeur de dimension n") ;
Lire(n) ;
Jusqu'à ( $n > 0$ ) et ( $n \leq 20$ ) ;
{Lecture des éléments de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
Lire(A[i][j]) ;
Fait ;
Fait ;
som ← 0 ;
Pour i allant de 1 à n pas 1
Faire
som ← som + A[i][i] ;
Fait ;
{Affichage de la somme}
Ecrire ("La somme des éléments de la diagonale principale de A est ", som) ;
Fin.
```

Exercice 12: Ecrire un algorithme qui transpose une matrice A carrée de caractères et de dimension n ($n \leq 10$). **(Devoir)**

Solution :

➤ Algorithme :

```
Algorithme exercice12 ;
Variables A, TA : tableau[10,10] entier ;
          n,i, som : entier ;
Début
{Lecture de la dimension et vérification n ( $n > 0$  et  $n \leq 10$ )}
Répéter
Ecrire ("Donner la valeur de dimension n") ;
Lire(n) ;
Jusqu'à ( $n > 0$ ) et ( $n \leq 10$ ) ;
{Lecture des éléments de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
Lire(A[i][j]) ;
Fait ;
Fait ;
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
TA[i][j] ← A[j][i] ;
Fait ;
Fait ;
{Affichage de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
Ecrire ("L'élément de ligne i=", i, " et colonne j=", j, " de la matrice A est ", A[i][j]) ;
Fait ;
Fait ;
Affichage de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
Ecrire ("L'élément de ligne i=", i, " et colonne j=", j, " de la matrice transposée TA est ", TA[i][j]) ;
Fait ;
Fait ;
Fin.
```

Exercice 13: Ecrire un algorithme qui permute les éléments des deux diagonales principale et secondaire d'une matrice A carrée d'entiers et de dimension n ($n \leq 50$).

Solution :

➤ Algorithme :

```
Algorithme exercice13 ;
Variables A : tableau[50, 50] entier ;
        n, i, X : entier ;
Début
{Lecture de la dimension et vérification n (n>0 et n≤50)}
Répéter
Ecrire ("Donner la valeur de dimension n") ;
Lire(n) ;
Jusqu'à (n>0) et (n≤50) ;
{Lecture des éléments de la matrice A}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à m pas 1
Faire
Ecrire ("Donner la valeur de l'élément de ligne i=", i, " et colonne j=", j) ;
Lire(A[i][j]) ;
Fait ;
Fait ;
Pour i allant de 1 à n pas 1
Faire
X ← A[i][i];
A[i][i] ← A[i][n-i+1];
A[i][n-i+1] ← X;
Fait ;
{Affichage des éléments de la matrice A après permutation des deux diagonales principale et secondaire}
Pour i allant de 1 à n pas 1
Faire
Pour j allant de 1 à n pas 1
Faire
Ecrire ("L'élément de ligne i=", i, " et colonne j=", j, "est", A[i][j]) ;
Fait ;
Fait ;
Fin.
```