

**BREVET DE TECHNICIEN SUPERIEUR  
DEVELOPPEMENT DES SYSTEMES D'INFORMATION**



## **Programmation en C**

**PLUS DE 150 EXERCICES AVEC  
CORRECTIONS**

Professeur : ELMOUKHTAR Noureddine

**ANNEE UNIVERSITAIRE : 2019-2020**

## Sommaire :

<b>ENONCES DES EXERCICES.....</b>	<b>9</b>
1. LES VARIABLES ET LES OPERATEURS .....	10
1-1) <i>Exercice 1: Opérateurs arithmétiques .....</i>	10
1-2) <i>Exercice 2 : Opérateurs de comparaison .....</i>	10
1-3) <i>Exercice 3 : Opérateurs logiques .....</i>	10
1-4) <i>Exercice 4 : Opérateurs de manipulation des bits .....</i>	11
1-5) <i>Exercice 5 : Le casting.....</i>	11
1-6) <i>Exercice 6 : Priorité des opérateurs arithmétiques.....</i>	11
1-7) <i>Exercice 7-1 : Echange de deux variables.....</i>	12
1-8) <i>Exercice 8-1 : Permutation de 3 variables .....</i>	12
2. LES ENTREES SORTIES CONVERSATIONNELLES .....	13
2-1) <i>Exercice 1 : Affichage avec retour à la ligne.....</i>	13
2-2) <i>Exercice 2 : Affichage des entiers .....</i>	13
2-3) <i>Exercice 3 : La somme de deux entiers .....</i>	13
2-4) <i>Exercice 4 : Le produit de deux entiers .....</i>	13
2-5) <i>Exercice 5 : Le quotient et le reste de deux entiers.....</i>	13
2-6) <i>Exercice 6: La division de deux nombres .....</i>	14
2-7) <i>Exercice 7 : L'échange avec affichage de résultat .....</i>	14
2-8) <i>Exercice 8 : La saisie des caractères .....</i>	14
2-9) <i>Exercice 9: La permutation avec affichage du résultat .....</i>	14
2-10) <i>Exercice 10 : La taille des types primitifs .....</i>	15
2-11) <i>Exercice 11 : Le code ASCII d'un caractère .....</i>	15
2-12) <i>Exercice 12 : Conversion d'une durée en secondes.....</i>	15
2-13) <i>Exercice 13 : Conversion d'une durée en (H : Min : S) .....</i>	15
2-14) <i>Exercice 14 : L'unité d'un entier .....</i>	16
2-15) <i>Exercice 15 : Conversion minuscule - majuscule.....</i>	16
2-16) <i>Exercice 16 : La surface d'un cercle .....</i>	16
2-17) <i>Exercice 17 : Le volume et la surface d'une sphère .....</i>	16
2-18) <i>Exercice 18 : La surface totale et le volume d'un cylindre .....</i>	17
2-19) <i>Exercice 19 : La moyenne de 3 nombres.....</i>	17
2-20) <i>Exercice 20 : Le prix TTC .....</i>	17
3. LES INSTRUCTIONS DE CONTROLE : IF ET SWITCH.....	18
3-1) <i>Exercice 1 : Pair ou impair.....</i>	18
3-2) <i>Exercice 2 : Positif, négatif ou nul .....</i>	18
3-3) <i>Exercice 3 : Multiple d'un entier ou non .....</i>	18
3-4) <i>Exercice 4 : Diviseur d'un entier ou non.....</i>	18
3-5) <i>Exercice 5 : Le maximum de deux nombres.....</i>	18
3-6) <i>Exercice 6 : Le minimum de 3 nombres .....</i>	19
3-7) <i>Exercice 7 : La valeur absolue.....</i>	19
3-8) <i>Exercice 8 : Signe d'un produit de deux nombres .....</i>	19
3-9) <i>Exercice 9 : La mention d'un stagiaire.....</i>	19
3-10) <i>Exercice 10 : L'état de l'eau.....</i>	20
3-11) <i>Exercice 11 : L'inverse d'un nombre .....</i>	20
3-12) <i>Exercice 12 : Type d'un triangle.....</i>	20
3-13) <i>Exercice 13 : Plus une minute .....</i>	20

3-14)	<i>Exercice 14 : Plus une seconde .....</i>	21
3-15)	<i>Exercice 15 : Moins une minute.....</i>	21
3-16)	<i>Exercice 16 : Moins une seconde.....</i>	21
3-17)	<i>Exercice 17 : La durée du voyage .....</i>	21
3-18)	<i>Exercice 18 : L'heure d'arrivée.....</i>	21
3-19)	<i>Exercice 19 : Année bissextile ou non.....</i>	22
3-20)	<i>Exercice 20 : Plus un jour.....</i>	22
3-21)	<i>Exercice 21 : Moins un jour .....</i>	22
3-22)	<i>Exercice 22 : Equation de 1<sup>er</sup> degré .....</i>	22
3-23)	<i>Exercice 23 : Equation de 2<sup>ème</sup> degré .....</i>	23
3-24)	<i>Exercice 24 : Une petite calculatrice.....</i>	23
4.	<b>LES INSTRUCTIONS DE CONTROLE – LES BOUCLES (FOR) .....</b>	24
4-1)	<i>Exercice 1 : Affichage d'un message plusieurs fois.....</i>	24
4-2)	<i>Exercice 2 : Affichage d'une suite de nombres .....</i>	24
4-3)	<i>Exercice 3 : Affichage avec condition .....</i>	24
4-4)	<i>Exercice 4 : La saisie de plusieurs nombres .....</i>	25
4-5)	<i>Exercice 5 : Calcul d'une somme S1 .....</i>	25
4-6)	<i>Exercice 6 : Calcul d'une somme S2 .....</i>	25
4-7)	<i>Exercice 7 : Calcul d'une somme S3 .....</i>	25
4-8)	<i>Exercice 8 : Calcul d'une somme S4 .....</i>	25
4-9)	<i>Exercice 9 : Calcul d'une somme S5 .....</i>	26
4-10)	<i>Exercice 10 : Calcul d'un produit P1 .....</i>	26
4-11)	<i>Exercice 11 : Calcul d'un produit P2 .....</i>	26
4-12)	<i>Exercice 12 : Les puissances de 2 .....</i>	26
4-13)	<i>Exercice 13 : Les diviseurs d'un entier .....</i>	26
4-14)	<i>Exercice 14 : Nombre premier ou non .....</i>	27
4-15)	<i>Exercice 15 : La puissance a<sup>n</sup> .....</i>	27
4-16)	<i>Exercice 16 : Le maximum de plusieurs nombres .....</i>	27
4-17)	<i>Exercice 17 : Terme d'une suite U<sub>n</sub> .....</i>	27
4-18)	<i>Exercice 18 : Terme d'une suite U<sub>n</sub> (2).....</i>	28
4-19)	<i>Exercice 19 : Affichage d'une forme (1).....</i>	28
4-20)	<i>Exercice 20 : Affichage d'une forme (2).....</i>	28
5.	<b>LES INSTRUCTIONS DE CONTROLE – LES BOUCLES (WHILE–DO WHILE) .....</b>	29
5-1)	<i>Exercice 1 : Conversion for ➔ while .....</i>	29
5-2)	<i>Exercice 2 : Contrôle de saisie (condition simple).....</i>	29
5-3)	<i>Exercice 3 : Contrôle de saisie (condition composée) .....</i>	29
5-4)	<i>Exercice 4 : Encadrement d'un nombre par puissance de 2 .....</i>	29
5-5)	<i>Exercice 5 : Le premier multiple supérieur d'un nombre n .....</i>	30
5-6)	<i>Exercice 6 : Nombre premier ou non .....</i>	30
5-7)	<i>Exercice 7 : Somme des nombres positifs seulement .....</i>	30
5-8)	<i>Exercice 8 : Le nombre d'apparition d'un nombre positif.....</i>	30
5-9)	<i>Exercice 9 : Le PGCD .....</i>	31
5-10)	<i>Exercice 10 : Le quotient et le reste sans utiliser / et %.....</i>	31
5-11)	<i>Exercice 11 : Nombres premiers entre eux .....</i>	31
5-12)	<i>Exercice 12 : Un nombre est en octal ou non .....</i>	31
5-13)	<i>Exercice 13 : Un nombre est en binaire ou non .....</i>	31
5-14)	<i>Exercice 14 : Conversion décimale - octale.....</i>	32
5-15)	<i>Exercice 15 : Conversion décimale - binaire .....</i>	32

5-16) Exercice 16 : Conversion binair.....	32
5-17) Exercice 17 : Conversion octale - décimale.....	32
6. LA PROGRAMMATION MODULAIRE : LES FONCTIONS .....	33
6-1) Exercice 1 : Fonction produit .....	33
6-2) Exercice 2 : Fonction surface d'un cercle.....	33
6-3) Exercice 3 : Fonction échange .....	33
6-4) Exercice 4 : Fonction si pair ou non .....	33
6-5) Exercice 5 : Fonction signe .....	33
6-6) Exercice 6 : Fonction si multiple ou non .....	34
6-7) Exercice 7 : Fonction si diviseur ou non .....	34
6-8) Exercice 8 : Fonction maximum.....	34
6-9) Exercice 9 : Fonction inverse.....	34
6-10) Exercice 10 : Fonction Message n fois .....	34
6-11) Exercice 11 : Fonction somme S1 .....	34
6-12) Exercice 12 : Fonction Factorielle .....	35
6-13) Exercice 13 : Fonction si premier ou non .....	35
6-14) Exercice 14 : Fonction puissance .....	35
6-15) Exercice 15 : Fonction qui affiche une forme.....	35
6-16) Exercice 16 : Fonction de contrôle de saisie .....	35
6-17) Exercice 17 : Fonction PGCD.....	36
6-18) Exercice 18 : Fonction si octal ou non.....	36
6-19) Exercice 19 : Fonction si binaire ou non .....	36
6-20) Exercice 20 : Fonction conversion décimale-octale .....	36
6-21) Exercice 21 : Fonction conversion décimale-binaire .....	36
6-22) Exercice 22 : Récursivité .....	36
7. LES TABLEAUX .....	37
7-1) Exercice 1 : Saisie et affichage d'un tableau .....	37
7-2) Exercice 2 : Calcul de la valeur maximale d'un tableau.....	37
7-3) Exercice 3 : Recherche dans un tableau .....	37
7-4) Exercice 4 : Affichage filtré des éléments d'un tableau.....	37
7-5) Exercice 5 : Le nombre d'apparition dans un tableau .....	37
7-6) Exercice 6* : Conversion décimale - binaire .....	37
7-7) Exercice 7 : Inverser les éléments d'un tableau.....	38
7-8) Exercice 8 : Changer certaines valeurs d'un tableau .....	38
7-9) Exercice 9 : La somme et la moyenne d'un tableau .....	38
7-10) Exercice 10 : Suppression des cases d'un tableau .....	38
7-11) Exercice 11 : Tri à bulle croissant .....	38
7-12) Exercice 12 : Tri à bulle décroissant.....	39
7-13) Exercice 13 : Tri par sélection croissant.....	39
7-14) Exercice 14 : Tri par sélection décroissant.....	39
7-15) Exercice 15 : Manipulation des notes des stagiaires .....	39
7-16) Exercice 16 : Tableau à deux dimensions .....	39
7-17) Exercice 17 : La trace d'une matrice .....	40
8. LES CHAINES DE CARACTERES.....	41
8-1) Exercice 1: Saisie et affichage d'une chaine .....	41
8-2) Exercice 2 : Copie d'une chaine .....	41
8-3) Exercice 3 : Affichage d'une partie d'une chaine.....	41
8-4) Exercice 4 : Affichage par ordre alphabétique de deux chaines .....	41

8-5)	<i>Exercice 5 : Ignorance de la casse lors de la comparaison de deux chaînes .....</i>	41
8-6)	<i>Exercice 6 : Concaténation de deux chaînes.....</i>	42
8-7)	<i>Exercice 7 : Chaine à l'envers .....</i>	42
8-8)	<i>Exercice 8 : Chaine palindrome .....</i>	42
8-9)	<i>Exercice 9 *: Nombre de mot d'une chaine .....</i>	42
8-10)	<i>Exercice 10 *: Suppression d'un caractère de toute la chaine.....</i>	42
8-11)	<i>Exercice 11 : Conversion minuscule-majuscule .....</i>	43
8-12)	<i>Exercice 12 : Remplacer une chaine par une autre dans une chaine.....</i>	43
8-13)	<i>Exercice 13 : Chiffrement et déchiffrement d'une chaine.....</i>	43
9.	<b>LES STRUCTURES .....</b>	44
9-1)	<i>Exercice 1 : La manipulation d'un nombre complexe .....</i>	44
9-2)	<i>Exercice 2 : La manipulation d'un nombre rationnel.....</i>	44
9-3)	<i>Exercice 3 : La manipulation d'une structure stagiaire .....</i>	45
9-4)	<i>Exercice 4 : La manipulation d'une structure « heure » .....</i>	45
9-5)	<i>Exercice 5 : La manipulation d'une structure « date » .....</i>	46
9-6)	<i>Exercice 6 : La manipulation d'une adresse IPv4.....</i>	46
10.	<b>LES FICHIERS .....</b>	48
10-1)	<i>Exercice 1 : Lecture et écriture dans un fichier .....</i>	48
10-2)	<i>Exercice 2 : Ecriture avec sauvegarde du contenu .....</i>	48
10-3)	<i>Exercice 3 : Ecriture en boucle.....</i>	48
10-4)	<i>Exercice 4 : La taille d'un fichier .....</i>	48
10-5)	<i>Exercice 5 : Recherche d'un caractère et son nombre d'apparition .....</i>	48
10-6)	<i>Exercice 6 : Recherche d'une chaîne de caractères et son nombre d'apparition.....</i>	48
10-7)	<i>Exercice 7 : Nombre de lignes d'un fichier.....</i>	49
10-8)	<i>Exercice 8 : Affichage filtré d'une partie du fichier.....</i>	49
10-9)	<i>Exercice 9 : Remplacer un caractère dans un fichier .....</i>	49
10-10)	<i>Exercice 10 : Remplacer une chaine dans un fichier .....</i>	49
10-11)	<i>Exercice 11 : Supprimer un caractère d'un fichier.....</i>	49
10-12)	<i>Exercice 12 : Supprimer une chaine d'un fichier .....</i>	49
	<b>CORRIGES DES EXERCICES.....</b>	50
1.	<b>LES VARIABLES ET LES OPERATEURS : .....</b>	51
1-1)	<i>Exercice 1-1 : .....</i>	51
1-2)	<i>Exercice 2-1 : .....</i>	51
1-3)	<i>Exercice 3-1 : .....</i>	51
1-4)	<i>Exercice 4-1 : .....</i>	52
1-5)	<i>Exercice 5-1 : .....</i>	53
1-6)	<i>Exercice 6-1 : .....</i>	53
1-7)	<i>Exercice 7-1 : .....</i>	53
1-8)	<i>Exercice 8-1 : .....</i>	54
2.	<b>LES INSTRUCTIONS D'ENTREES SORTIES CONVERSATIONNELLES .....</b>	55
2-1)	<i>Exercice 1-2 : .....</i>	55
2-2)	<i>Exercice 2-2 : .....</i>	55
2-3)	<i>Exercice 3-2 : .....</i>	55
2-4)	<i>Exercice 4-2 : .....</i>	56
2-5)	<i>Exercice 5-2 : .....</i>	56
2-6)	<i>Exercice 6-2 : .....</i>	57
2-7)	<i>Exercice 7-2 : .....</i>	57

2-8)	<i>Exercice 8-2 :</i>	58
2-9)	<i>Exercice 9-2 :</i>	59
2-10)	<i>Exercice 10-2 :</i>	59
2-11)	<i>Exercice 11-2 :</i>	59
2-12)	<i>Exercice 12-2 :</i>	60
2-13)	<i>Exercice 13-2 :</i>	60
2-14)	<i>Exercice 14-2 :</i>	60
2-15)	<i>Exercice 15-2 :</i>	61
2-16)	<i>Exercice 16-2 :</i>	61
2-17)	<i>Exercice 17-2 :</i>	61
2-18)	<i>Exercice 18-2 :</i>	62
2-19)	<i>Exercice 19-2 :</i>	62
2-20)	<i>Exercice 20-2 :</i>	62
3.	LES INSTRUCTIONS DE CONTROLE – IF ET SWITCH	63
3-1)	<i>Exercice 1-3 :</i>	63
3-2)	<i>Exercice 2-3 :</i>	63
3-3)	<i>Exercice 3-3 :</i>	63
3-4)	<i>Exercice 4-3 :</i>	64
3-5)	<i>Exercice 5-3 :</i>	64
3-6)	<i>Exercice 6-3 :</i>	64
3-7)	<i>Exercice 7-3 :</i>	65
3-8)	<i>Exercice 8-3 :</i>	65
3-9)	<i>Exercice 9-3 :</i>	65
3-10)	<i>Exercice 10-3 :</i>	66
3-11)	<i>Exercice 11-3 :</i>	66
3-12)	<i>Exercice 12-3 :</i>	66
3-13)	<i>Exercice 13-3 :</i>	67
3-14)	<i>Exercice 14-3 :</i>	68
3-15)	<i>Exercice 15-3 :</i>	69
3-16)	<i>Exercice 16-3 :</i>	69
3-17)	<i>Exercice 17-3 :</i>	70
3-18)	<i>Exercice 18-3 :</i>	71
3-19)	<i>Exercice 19-3 :</i>	71
3-20)	<i>Exercice 20-3 :</i>	72
3-21)	<i>Exercice 21-3 :</i>	73
3-22)	<i>Exercice 22-3 :</i>	75
3-23)	<i>Exercice 23-3 :</i>	75
3-24)	<i>Exercice 24-3 :</i>	76
4.	LES INSTRUCTIONS DE CONTROLE – LES BOUCLES (FOR)	77
4-1)	<i>Exercice 1-4 :</i>	77
4-2)	<i>Exercice 2-4 :</i>	77
4-3)	<i>Exercice 3-4 :</i>	77
4-4)	<i>Exercice 4-4 :</i>	78
4-5)	<i>Exercice 5-4 :</i>	78
4-6)	<i>Exercice 6-4 :</i>	78
4-7)	<i>Exercice 7-4 :</i>	78
4-8)	<i>Exercice 8-4 :</i>	79
4-9)	<i>Exercice 9-4 :</i>	79

4-10)	<i>Exercice 10-4 :</i>	79
4-11)	<i>Exercice 11-4 :</i>	80
4-12)	<i>Exercice 12-4 :</i>	80
4-13)	<i>Exercice 13-4 :</i>	80
4-14)	<i>Exercice 14-4 :</i>	81
4-15)	<i>Exercice 15-4 :</i>	81
4-16)	<i>Exercice 16-4 :</i>	81
4-17)	<i>Exercice 17-4 :</i>	82
4-18)	<i>Exercice 18-4 :</i>	82
4-19)	<i>Exercice 19-4 :</i>	82
4-20)	<i>Exercice 20-4 :</i>	83
5.	LES INSTRUCTIONS DE CONTROLE – LES BOUCLES (WHILE ET DO WHILE) .....	84
5-1)	<i>Exercice 1-5 :</i>	84
5-2)	<i>Exercice 2-5 :</i>	84
5-3)	<i>Exercice 3-5 :</i>	84
5-4)	<i>Exercice 4-5 :</i>	85
5-5)	<i>Exercice 5-5 :</i>	85
5-6)	<i>Exercice 6-5 :</i>	85
5-7)	<i>Exercice 7-5 :</i>	86
5-8)	<i>Exercice 8-5 :</i>	86
5-9)	<i>Exercice 9-5 :</i>	86
5-10)	<i>Exercice 10-5 :</i>	87
5-11)	<i>Exercice 11-5 :</i>	87
5-12)	<i>Exercice 12-5 :</i>	88
5-13)	<i>Exercice 13-5 :</i>	88
5-14)	<i>Exercice 14-5 :</i>	89
5-15)	<i>Exercice 15-5 :</i>	89
5-16)	<i>Exercice 16-5 :</i>	89
5-17)	<i>Exercice 17-5 :</i>	90
6.	LA PROGRAMMATION MODULAIRE : LES FONCTIONS .....	91
6-1)	<i>Exercice 1-6 :</i>	91
6-2)	<i>Exercice 2-6 :</i>	91
6-3)	<i>Exercice 3-6 :</i>	92
6-4)	<i>Exercice 4-6 :</i>	92
6-5)	<i>Exercice 5-6 :</i>	93
6-6)	<i>Exercice 6-6 :</i>	93
6-7)	<i>Exercice 7-6 :</i>	94
6-8)	<i>Exercice 8-6 :</i>	94
6-9)	<i>Exercice 9-6 :</i>	95
6-10)	<i>Exercice 10-6 :</i>	95
6-11)	<i>Exercice 11-6 :</i>	96
6-12)	<i>Exercice 12-6 :</i>	96
6-13)	<i>Exercice 13-6 :</i>	97
6-14)	<i>Exercice 14-6 :</i>	97
6-15)	<i>Exercice 15-6 :</i>	98
6-16)	<i>Exercice 16-6 :</i>	98
6-17)	<i>Exercice 17-6 :</i>	98
6-18)	<i>Exercice 18-6 :</i>	99

6-19)	<i>Exercice 19-6 :</i>	99
6-20)	<i>Exercice 20-6 :</i>	100
6-21)	<i>Exercice 21-6 :</i>	100
6-22)	<i>Exercice 22-6 :</i>	101
7.	LES TABLEAUX .....	102
7-1)	<i>Exercice 1-7 :</i>	102
7-2)	<i>Exercice 2-7 :</i>	102
7-3)	<i>Exercice 3-7 :</i>	103
7-4)	<i>Exercice 4-7 :</i>	103
7-5)	<i>Exercice 5-7 :</i>	104
7-6)	<i>Exercice 6-7 :</i>	104
7-7)	<i>Exercice 7-7 :</i>	105
7-8)	<i>Exercice 8-7 :</i>	106
7-9)	<i>Exercice 9-7 :</i>	107
7-10)	<i>Exercice 10-7 :</i>	107
7-11)	<i>Exercice 11-7 :</i>	108
7-12)	<i>Exercice 12-7 :</i>	108
7-13)	<i>Exercice 13-7 :</i>	109
7-14)	<i>Exercice 14-7 :</i>	110
7-15)	<i>Exercice 15-7 :</i>	111
7-16)	<i>Exercice 16-7 :</i>	112
7-17)	<i>Exercice 17-7 :</i>	112
8.	LES CHAINES DE CARACTERES.....	113
8-1)	<i>Exercice 1-8 :</i>	113
8-2)	<i>Exercice 2-8 :</i>	113
8-3)	<i>Exercice 3-8 :</i>	113
8-4)	<i>Exercice 4-8 :</i>	114
8-5)	<i>Exercice 5-8 :</i>	114
8-6)	<i>Exercice 6-8 :</i>	115
8-7)	<i>Exercice 7-8 :</i>	115
8-8)	<i>Exercice 8-8 :</i>	115
8-9)	<i>Exercice 9-8 :</i>	116
8-10)	<i>Exercice 10-8 :</i>	116
8-11)	<i>Exercice 11-8 :</i>	117
8-12)	<i>Exercice 12-8 :</i>	117
8-13)	<i>Exercice 13-8 :</i>	118
9.	LES STRUCTURES .....	119
9-1)	<i>Exercice 1-9 :</i>	119
9-2)	<i>Exercice 2-9 :</i>	120
9-3)	<i>Exercice 3-9 :</i>	122
9-4)	<i>Exercice 4-9 :</i>	124
9-5)	<i>Exercice 5-9 :</i>	127
9-6)	<i>Exercice 6-9 :</i>	130
10.	LES FICHIERS .....	134
10-1)	<i>Exercice 1-10 :</i>	134
10-2)	<i>Exercice 2-10 :</i>	134
10-3)	<i>Exercice 3-10 :</i>	135
10-4)	<i>Exercice 4-10 :</i>	136

---

10-5) <i>Exercice 5-10 :</i> .....	136
10-6) <i>Exercice 6-10 :</i> .....	137
10-7) <i>Exercice 7-10 :</i> .....	137
10-8) <i>Exercice 8-10 :</i> .....	138
10-9) <i>Exercice 9-10 :</i> .....	139
10-10) <i>Exercice 10-10 :</i> .....	140
10-11) <i>Exercice 11-10 :</i> .....	141
10-12) <i>Exercice 12-10 :</i> .....	142

## Enoncés des exercices



# ENONCÉS DES EXERCICES

## 1. Les variables et les opérateurs

### 1-1) Exercice 1: Opérateurs arithmétiques

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
#include <stdio.h>
main (){
    int a=15,b=3,c,d;
    a=a-b;
    b=a/b;
    c=a+b*4;
    d=c%4;
    printf("%d %d %d %d",a,b,c,d);
}
```

### 1-2) Exercice 2 : Opérateurs de comparaison

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
#include <stdio.h>
main (){
    int a=15,b=3,c,d;
    a=(a>=b);
    b=(a==b);
    c=(a!=b);
    d=(a=5);
    printf("%d %d %d %d",a,b,c,d);
}
```

### 1-3) Exercice 3 : Opérateurs logiques

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
main ()
{
    int a=15,b=3,c,d;
    a= (b!=a) && (b>2);
    b= (b=4) || (a>5);
    c= !(a<=15);
    d=! (c==0) && (b!=a);
    printf("%d %d %d %d",a,b,c,d);
}
```

### 1-4) Exercice 4 : opérateurs de manipulation des bits

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
#include <stdio.h>
main ()
{
    int a=15,b=3,c,d;
    a= a & 10;
    b= 14 | b;
    c= a & b;
    d= c ^ b;
    printf("%d %d %d %d",a,b,c,d);
}
```

### 1-5) Exercice 5 : Le casting

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
#include <stdio.h>
main ()
{
    float d,e,f;
    int a=20, b=3,c;
    c=a/b;
    d=a/b;
    e=(float)a/b;
    f=(float)(a/b);
    printf("%d %d %d %f %f %f",a,b,c,d,e,f);
}
```

### 1-6) Exercice 6 : Priorité des opérateurs arithmétiques

Quelles sont les valeurs des variables après l'exécution de ce programme ?

```
#include <stdio.h>
main () {
    int a=10, b=7,c,d,e;
    c=a/2*b;
    d=b/3*c;
    e=c+d/2;
    printf("%d %d %d %d %d",a,b,c,d,e);
}
```

**1-7) Exercice 7-1 : échange de deux variables**

Ecrire un programme qui permet d'échanger le contenu de deux variables entières.

**Exemple :**

	A	b
Avant l'échange	5	3
Après l'échange	3	5

**1-8) Exercice 8-1 : Permutation de 3 variables**

Ecrire un programme qui permet de permuter le contenu de trois variables entières avec la permutation suivante  $f(a,b,c) = (c,a,b)$ .

**Exemple :**

	A	B	c
Avant la permutation	5	3	7
Après la permutation	7	5	3

## 2. Les entrées sorties conversationnelles

### 2-1) Exercice 1 : Affichage avec retour à la ligne

Ecrire un programme qui permet d'afficher le message suivant :

**Résultat d'exécution :**

Bonjour Monsieur

Merci

### 2-2) Exercice 2 : Affichage des entiers

Ecrire un programme qui permet d'afficher le message suivant (les nombres sont séparés par des tabulations et chaque nombre occupe 3 positions).

**Résultat d'exécution :**

5      10     150      7

120      1      14      78

### 2-3) Exercice 3 : La somme de deux entiers

Ecrire un programme qui demande à l'utilisateur de saisir deux entiers et qui affiche leur somme.

**Exemple d'exécution :**

Saisir un entier a : 5

Saisir un entier b : 7

La somme est : 12

### 2-4) Exercice 4 : Le produit de deux entiers

Ecrire un programme qui demande à l'utilisateur de saisir deux entiers et qui affiche leur produit.

**Exemple d'exécution :**

Saisir un entier a : 5

Saisir un entier b : 7

Le produit est : 35

### 2-5) Exercice 5 : Le quotient et le reste de deux entiers

Ecrire un programme qui demande à l'utilisateur de saisir deux entiers a et b et qui affiche le quotient et le reste de la division Euclidienne de a par b.

**Exemple d'exécution :**

Saisir un entier a : 14

Saisir un entier b : 5

Le quotient est : 2

Le reste est : 4

## 2-6) Exercice 6: La division de deux nombres

Ecrire un programme qui demande à l'utilisateur de saisir deux entiers a et b et qui affiche le résultat de leur division exacte (deux chiffres après la virgule).

**Exemple d'exécution :**

Saisir un entier a : 5

Saisir un entier b : 2

5 / 2 = 2.50

## 2-7) Exercice 7 : L'échange avec affichage de résultat

Ecrire un programme qui demande à l'utilisateur de :

- Saisir deux entiers.
- Affiche les deux entiers.
- Echange leurs valeurs.
- Affiche les deux entiers après l'échange.

**Exemple d'exécution :**

Saisir un entier a : 5

Saisir un entier b : 7

Avant :

a = 5        b = 7

Après :

a = 7        b = 5

## 2-8) Exercice 8 : La saisie des caractères

Ecrire un programme qui demande à l'utilisateur de saisir trois caractères et qui les affiche.

- Utiliser *printf* et *scanf* dans la première version.
- Utiliser *getchar* et *putchar* dans la deuxième version.

**Exemple d'exécution :**

Saisir un caractère a : 5

Saisir un caractère b : K

Saisir un caractère c : !

a = 5

b = K

c = !

## 2-9) Exercice 9: La permutation avec affichage du résultat

Ecrire un programme qui demande à l'utilisateur de saisir trois entiers et qui les affiche après la permutation suivante  $F(a, b, c) = (b, c, a)$ .

**Exemple d'exécution :**

Saisir un entier a : 5

Saisir un entier b : 7

Saisir un entier b : 3

Avant :

a = 5        b = 7        c = 3

Après :

a = 7        b = 3        c = 5

### **2-10)      Exercice 10 : La taille des types primitifs**

Ecrire un programme qui permet d'afficher la taille des types primaires en octet (*int*, *float*, *char*).

**Exemple d'exécution :**

Taille d'un entier (*int*) : 4 octet(s)

Taille d'un réel (*float*) : 4 octet(s)

Taille d'un caractère (*char*) : 1 octet(s)

### **2-11)      Exercice 11 : Le code ASCI d'un caractère**

Ecrire un programme qui permet d'afficher le code ASCI d'un caractère saisi par l'utilisateur.

**Exemple d'exécution :**

Saisir un caractère : A

Le code ASCI de (A) est : 65

### **2-12)      Exercice 12 : Conversion d'une durée en secondes**

Ecrire un programme qui demande à l'utilisateur de saisir une durée en (HH : MM :SS) et qui l'affiche en secondes.

**Exemple d'exécution :**

Saisir une durée en (HH :MM :SS) : 1 :5 :20

La durée en secondes est : 3920 Secondes

### **2-13)      Exercice 13 : Conversion d'une durée en (H : Min : S)**

Ecrire un programme qui demande à l'utilisateur de saisir une durée en en secondes et qui l'affiche en (HH : MM :SS).

**Exemple d'exécution :**

Saisir la durée en secondes : 3920 Secondes

La durée en (HH :MM :SS, . . . . .)

### **2-14) Exercice 14 : L'unité d'un entier**

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche la valeur de son unité

**Exemple d'exécution :**

Saisir un entier a : 128

Son unité est : 8

### **2-15) Exercice 15 : Conversion minuscule - majuscule**

Ecrire un programme qui demande à l'utilisateur de saisir un caractère en minuscule et qui l'affiche en majuscule.

**Exemple d'exécution :**

Saisir un caractère en minuscule: h

Le caractère en majuscule est : H

### **2-16) Exercice 16 : La surface d'un cercle**

Ecrire un programme qui demande à l'utilisateur de saisir le rayon d'un cercle et qui permet de calculer sa surface. La surface est  $S=\pi R^2$

**Exemple d'exécution :**

Saisir le rayon d'un cercle : 5

La surface du cercle est : 78.50

### **2-17) Exercice 17 : Le volume et la surface d'une sphère**

Ecrire un programme qui demande à l'utilisateur de saisir le rayon d'une sphère et qui permet de calculer sa surface et son volume.

La surface d'une sphère de rayon R est  $S=4\pi R^2$

Le volume d'une sphère de rayon R est  $V= (4/3) \pi R^3$

**Exemple d'exécution :**

Saisir le rayon d'une sphère : 5

La surface de la sphère est : 314.00

Le volume de la sphère est : 392.50

### **2-18) Exercice 18 : La surface totale et le volume d'un cylindre**

Ecrire un programme qui demande à l'utilisateur de saisir le rayon et la hauteur d'un cylindre et qui permet de calculer sa surface totale et son volume.

La surface totale d'un cylindre de rayon R et de hauteur H est  $S=2\pi R (H+R)$   
Le volume d'un cylindre de rayon R et de hauteur H est  $V=\pi R^2 H$

**Exemple d'exécution :**

Saisir le rayon d'un cylindre : 5

Saisir la hauteur d'un cylindre : 10

La surface du cylindre est : 471.00

Le volume du cylindre : 785.00

### **2-19) Exercice 19 : La moyenne de 3 nombres**

Ecrire un programme qui demande à l'utilisateur de saisir 3 nombres et qui affiche leur moyenne.

La moyenne est la somme des trois nombres divisée par 3

**Exemple d'exécution :**

Saisir un nombre a : 5

Saisir un nombre b : 10

Saisir un nombre c : 6

La moyenne est : 7.00

### **2-20) Exercice 20 : Le prix TTC**

Ecrire un programme qui demande à l'utilisateur de saisir le prix d'un article en TTC et qui affiche sa valeur HT (le taux de TVA est 20%).

Prix toute taxe comprise =  $1.2 \times$  Prix hors taxe

**Exemple d'exécution :**

Saisir le prix hors taxe d'un article : 10

Le prix TTC : 12.00

### 3. Les instructions de contrôle : **if et switch**

#### 3-1) Exercice 1 : Pair ou impair

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche si cet entier est pair ou impair.

**Exemple d'exécution :**

Saisir un entier : 5  
5 est un nombre impair

#### 3-2) Exercice 2 : Positif, négatif ou nul

Ecrire un programme qui demande à l'utilisateur de saisir un nombre et qui affiche s'il est positif, négatif ou nul.

**Exemple d'exécution :**

Saisir un nombre : 5  
5.00 est un nombre positif

#### 3-3) Exercice 3 : Multiple d'un entier ou non

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche si cet entier est un multiple de 7 ou non.

**Exemple d'exécution :**

Saisir un entier : 49  
49 est un multiple de 7

#### 3-4) Exercice 4 : Diviseur d'un entier ou non

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche si cet entier est un diviseur de 100.

**Exemple d'exécution :**

Saisir un entier : 11  
11 n'est pas un diviseur de 100

#### 3-5) Exercice 5 : Le maximum de deux nombres

Ecrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche le maximum.

**Exemple d'exécution :**

Saisir un nombre a : 12.8  
Saisir un nombre b : -5.5  
Le maximum est : 12.8

### 3-6) Exercice 6 : Le minimum de 3 nombres

Ecrire un programme qui demande à l'utilisateur de saisir 3 nombres et qui affiche le minimum.

**Exemple d'exécution :**

Saisir un nombre a : 12.8

Saisir un nombre b : -5.5

Saisir un nombre c : -5.7

Le minimum est : -5.7

### 3-7) Exercice 7 : La valeur absolue

Ecrire un programme qui demande à l'utilisateur de saisir un nombre et qui affiche sa valeur absolue.

**Exemple d'exécution :**

Saisir un nombre a : -12.5

La valeur absolue de -12.5 est : 12.5

### 3-8) Exercice 8 : Signe d'un produit de deux nombres

Ecrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche si leur produit est positif, négatif ou nul sans calculer le produit

**Exemple d'exécution :**

Saisir un nombre : 5

Saisir un nombre : 8

Le produit est positif

### 3-9) Exercice 9 : La mention d'un stagiaire

Ecrire un programme qui demande à l'utilisateur de saisir la note d'un stagiaire et qui affiche sa mention.

Note	Mention
Note < 10	Echec
10<= Note <12	Passable
12<= Note <14	Assez bien
14<= Note <16	Bien
16<= Note <18	Très bien
18<= Note <=20	Excellent

**Exemple d'exécution :**

Saisir la note : 12.8

La mention est : Passable

**3-10) Exercice 10 : L'état de l'eau**

Ecrire un programme qui demande à l'utilisateur de saisir la température de l'eau et qui affiche son état physique.

Température	Etat
$T \leq 0$	Solide
$0 < T < 100$	Liquide
$T > 100$	Gaz

**Exemple d'exécution :**

Saisir la température : 12.8

L'eau est en état liquide

**3-11) Exercice 11 : L'inverse d'un nombre**

Ecrire un programme qui demande à l'utilisateur de saisir un nombre et qui affiche son inverse (Empêcher la division par 0).

**Exemple d'exécution :**

Saisir un nombre : 2

L'inverse est : 0.5

**3-12) Exercice 12 : Type d'un triangle**

Ecrire un programme qui demande à l'utilisateur de saisir les longueurs des côtés d'un triangle et qui affiche s'il est rectangle, isocèle, équilatéral ou non.

**Exemple d'exécution :**

Saisir les longueurs des côtés du triangle :

a = 2

b = 5

c = 2

Le triangle est :

Isocèle

**3-13) Exercice 13 : Plus une minute**

Ecrire un programme qui demande à l'utilisateur de saisir une heure (HH :MM) et qui affiche l'heure une minute après.

**Exemple d'exécution :**

Saisir une heure (HH :MM) : 15 :59

Nouvelle heure : 16 : 00

**3-14) Exercice 14 : Plus une seconde**

Ecrire un programme qui demande à l'utilisateur de saisir une heure (HH :MM :SS) et qui affiche l'heure une seconde après.

**Exemple d'exécution :**

Saisir une heure (HH :MM :SS) : 15 :59 :23  
Nouvelle heure : 15 :59 :24

**3-15) Exercice 15 : Moins une minute**

Ecrire un programme qui demande à l'utilisateur de saisir une heure (HH :MM) et qui affiche l'heure une minute avant.

**Exemple d'exécution :**

Saisir une heure (HH :MM) : 15 :59  
Nouvelle heure : 15 : 58

**3-16) Exercice 16 : Moins une seconde**

Ecrire un programme qui demande à l'utilisateur de saisir une heure (HH :MM :SS) et qui affiche l'heure une seconde avant.

**Exemple d'exécution :**

Saisir une heure (HH :MM :SS) : 15 :59 :23  
Nouvelle heure : 15 :59 :22

**3-17) Exercice 17 : La durée du voyage**

Ecrire un programme qui demande à l'utilisateur de saisir l'heure de départ du train (HH :MM) et l'heure d'arrivée (HH :MM) et qui affiche la durée du voyage.

**Exemple d'exécution :**

Saisir l'heure de départ (HH :MM) : 15 :20  
Saisir l'heure d'arrivée (HH :MM) : 16 :30  
La durée de voyage est : 01 :10

**3-18) Exercice 18 : L'heure d'arrivée**

Ecrire un programme qui demande à l'utilisateur de saisir l'heure de départ du train (HH :MM) et la durée de voyage en (HH :MM) et qui affiche l'heure d'arrivée.

**Exemple d'exécution :**

Saisir l'heure de départ (HH :MM) : 15 :20

Saisir la durée de voyage (HH :MM) : 01 :10

L'heure d'arrivée est : 16 :30

### **3-19) Exercice 19 : Année bissextile ou non**

Ecrire un programme qui demande à l'utilisateur de saisir une année et qui affiche si cette année est bissextile ou non.

**Exemple d'exécution :**

Saisir une année : 2017

Cette année est non bissextile

### **3-20) Exercice 20 : Plus un jour**

Ecrire un programme qui demande à l'utilisateur de saisir une date et qui affiche la date un jour après.

**Exemple d'exécution :**

Saisir une date (JJ/MM/AA) : 12/01/2017

La nouvelle date est : 13/01/2017

### **3-21) Exercice 21 : Moins un jour**

Ecrire un programme qui demande à l'utilisateur de saisir une date et qui affiche la date un jour avant.

**Exemple d'exécution :**

Saisir une date (JJ/MM/AA) : 12/01/2017

La nouvelle date est : 11/01/2017

### **3-22) Exercice 22 : Equation de 1<sup>er</sup> degré**

Ecrire un programme qui permet de résoudre une équation de 1<sup>er</sup> degré.

**Exemple d'exécution :**

Ce programme résout l'équation  $ax+b=0$

Saisir la valeur de a : 5

Saisir la valeur de b : 10

La solution est : -2

**3-23) Exercice 23 : équation de 2<sup>e</sup> degré**

Ecrire un programme qui permet de résoudre une équation de 2<sup>ème</sup> degré.

Vous allez utiliser la fonction `sqrt(x)` qui permet de calculer la racine carrée de x → Pour cela, importer la bibliothèque `math.h`

**Exemple d'exécution :**

Ce programme résout l'équation  $ax^2+bx+c=0$

Saisir la valeur de a : 1

Saisir la valeur de b : -2

Saisir la valeur de c : 1

x = 1

**3-24) Exercice 24 : Une petite calculatrice**

Ecrire un programme qui permet à l'utilisateur de choisir l'opération à effectuer (addition, soustraction, division ou multiplication) et qui affiche le résultat de calcul selon son choix.

**Exemple d'exécution :**

1 – Addition

2 – Soustraction

3 – Division

4 – Multiplication

Saisir la valeur de a : 5

Saisir la valeur de b : 2

Choisir l'opération : 4

Le résultat est : 10

#### 4. Les instructions de contrôle - Les boucles (for)

##### 4-1) Exercice 1 : Affichage d'un message plusieurs fois

Ecrire un programme qui permet d'afficher le message « Bonjour Mr QARA » 10 fois avec des retours à la ligne.

Résultat d'exécution :

```
Bonjour Mr QARA
```

##### 4-2) Exercice 2 : Affichage d'une suite de nombres

Ecrire un programme qui permet d'afficher les nombres entiers de 1 jusqu'à 10 avec des retours à la ligne.

Résultat d'exécution :

```
1
2
3
4
5
6
7
8
9
10
```

##### 4-3) Exercice 3 : Affichage avec condition

Ecrire un programme qui permet d'afficher les nombres pairs compris entre 1 et 20 avec des retours à la ligne.

Résultat d'exécution :

```
2
4
6
8
10
12
14
16
18
20
```

**4-4) Exercice 4 : La saisie de plusieurs nombres**

Ecrire un programme qui permet de saisir dix entiers.

**Résultat d'exécution :**

Saisir un entier : 15  
Saisir un entier : 10  
Saisir un entier : 7  
Saisir un entier : 8  
Saisir un entier : 98  
Saisir un entier : -45  
Saisir un entier : 5  
Saisir un entier : 14  
Saisir un entier : 12  
Saisir un entier : 10

**4-5) Exercice 5 : Calcul d'une somme S1**

Ecrire un programme qui permet de calculer la somme :

$$S1 = 1 + 2 + 3 + 4 + 5 + 6 + \dots + 100$$

**Résultat d'exécution :**

La somme S1 = 5050

**4-6) Exercice 6 : Calcul d'une somme S2**

Ecrire un programme qui permet de calculer la somme :

$$S2 = 1 + 3 + 5 + \dots + 99$$

**Résultat d'exécution :**

La somme S2 = 2500

**4-7) Exercice 7 : Calcul d'une somme S3**

Ecrire un programme qui permet de calculer la somme :

$$S3 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + \dots + 100^2$$

**Résultat d'exécution :**

La somme S3 = 338350

**4-8) Exercice 8 : Calcul d'une somme S4**

Ecrire un programme qui permet de calculer la somme :

$$S4 = 1/2 + 2/3 + 3/4 + 4/5 + 5/6 + \dots + 98/99 + 99/100$$

**Résultat d'exécution :**

La somme S4 = 94.81

**4-9) Exercice 9 : Calcul d'une somme S5**

Ecrire un programme qui permet de calculer la somme :

$$S5 = 1/2 - 2/3 + 3/4 - 4/5 + 5/6 - \dots - 98/99 + 99/100$$

Résultat d'exécution :

La somme S5 = 0.69

**4-10) Exercice 10 : Calcul d'un produit P1**

Ecrire un programme qui permet de calculer le produit :

$$P1 = 1 * 2 * 3 * 4 * 5 * 6 * \dots * 10$$

Résultat d'exécution :

Le produit = 3628800

**4-11) Exercice 11 : Calcul d'un produit P2**

Ecrire un programme qui permet de calculer le produit :

$$P2 = 1/2 * 3/4 * 5/6 * 7/8 * \dots * 19/20$$

Résultat d'exécution :

Le produit = 0.176197

**4-12) Exercice 12 : Les puissances de 2**

Ecrire un programme qui permet d'afficher les puissances de 2 jusqu'à  $2^{10}$ .

Résultat d'exécution :

$$\begin{aligned} 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &= 4 \\ 2^3 &= 8 \\ 2^4 &= 16 \\ 2^5 &= 32 \\ 2^6 &= 64 \\ 2^7 &= 128 \\ 2^8 &= 256 \\ 2^9 &= 512 \\ 2^{10} &= 1024 \end{aligned}$$

**4-13) Exercice 13 : Les diviseurs d'un entier**

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche tous ses diviseurs.

**Exemple d'exécution :**

Saisir un entier positif : 26

1        2        5        10        25

#### **4-14)      Exercice 14 : Nombre premier ou non**

Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche si ce nombre est premier ou non.

**Exemple d'exécution :**

Saisir un entier positif : 17

17 est premier

#### **4-15)      Exercice 15 : La puissance a<sup>n</sup>**

Ecrire un programme qui demande à l'utilisateur de saisir deux entier a et n et qui affiche la puissance a<sup>n</sup>.

**Exemple d'exécution :**

Saisir un entier a : 15

Saisir un entier positif n : 3

15<sup>3</sup> = 3375

#### **4-16)      Exercice 16 : Le maximum de plusieurs nombres**

Ecrire un programme qui permet de saisir dix entiers et qui affiche le maximum.

**Exemple d'exécution :**

Saisir dix entiers :

- 1) 8
- 2) 96
- 3) 48
- 4) 74
- 5) 10
- 6) 100
- 7) -45
- 8) -89
- 9) 10
- 10) -89

Le maximum est : 100

#### **4-17)      Exercice 17 : Terme d'une suite U<sub>n</sub>**

Ecrire un programme qui permet de calculer le n<sup>ième</sup> terme de la suite (n est saisi par l'utilisateur)

$$U_0 = 1$$

$$U_{n+1} = 2*U_n + 5$$

**Exemple d'exécution :**

Saisir un nombre : 15.5

$U(15)=196603$

#### **4-18)      Exercice 18 : Terme d'une suite $U_n$ (2)**

Ecrire un programme qui permet de calculer le  $n^{\text{ième}}$  terme de la suite  $U_n$  ( $n$  est saisi par l'utilisateur)

$$U_0 = 1$$

$$U_1 = 1$$

$$U_{n+2} = U_{n+1} + 2*U_n + 3$$

**Exemple d'exécution :**

Saisir un nombre : 5

$U(5)=51$

#### **4-19)      Exercice 19 : Affichage d'une forme (1)**

Ecrire un programme qui permet d'afficher les schémas suivants.

(Le nombre de lignes est saisi par l'utilisateur).

**Exemple d'exécution :**

+	+++++
++	++++
+++	+++
++++	++
+++++	+
<b>N=5</b>	<b>N=5</b>
<b>Schéma 1</b>	<b>Schéma 2</b>

#### **4-20)      Exercice 20 : Affichage d'une forme (2)**

Ecrire un programme qui permet d'afficher les schémas suivants.

(Le nombre de lignes est saisi par l'utilisateur).

**Exemple d'exécution :**

+	+++++
++	++++
+++	+++
++++	++
+++++	+
<b>N=5</b>	<b>N=5</b>
<b>Schéma 3</b>	<b>Schéma 4</b>

## 5. Les instructions de contrôle - Les boucles (while-do while)

### 5-1) Exercice 1 : Conversion for → while

Ecrire ce programme à l'aide des boucles while et do while

```
int i;  
  
for (i=1;i<=10;i++) printf ("Bonjour Mr QARA\n");
```

### 5-2) Exercice 2 : Contrôle de saisie (condition simple)

Ecrire un programme qui permet de saisir un nombre strictement positif jusqu'à ce que la réponse convienne.

**Exemple d'exécution :**

```
Saisir un nombre positif : -5  
Saisir un nombre positif : -8  
Saisir un nombre positif : 7
```

### 5-3) Exercice 3 : Contrôle de saisie (condition composée)

Ecrire un programme qui permet de saisir un nombre compris entre 0 et 20 jusqu'à ce que la réponse convienne.

Si l'utilisateur entre un nombre plus grand que 20, afficher lui « Plus petit » et si entre un nombre négatif, afficher lui « Plus grand ».

**Exemple d'exécution :**

```
Saisir un nombre : -5  
Plus grand  
Saisir un nombre : 30  
Plus petit  
Saisir un nombre : 7
```

### 5-4) Exercice 4 : Encadrement d'un nombre par puissance de 2

Ecrire un programme qui permet de saisir un entier positif n et qui affiche les puissances de deux les plus proches de n ( $2^x < n \leq 2^{x+1}$ )

**Exemple d'exécution :**

```
Saisir un entier positif : 17  
2^4 < 17 <= 2^5
```

### **5-5) Exercice 5 : Le premier multiple supérieur d'un nombre n**

Ecrire un programme qui permet de saisir un entier positif n et qui affiche le premier multiple de 5 supérieure à n.

**Exemple d'exécution :**

```
Saisir un entier positif : 17  
15 < 17 <= 20
```

### **5-6) Exercice 6 : Nombre premier ou non**

Ecrire un programme qui permet de saisir un entier positif et qui vérifie si ce nombre est premier ou non (Utiliser la boucle while pour optimiser le programme)

**Exemple d'exécution :**

```
Saisir un entier positif : 23  
23 est premier
```

### **5-7) Exercice 7 : Somme des nombres positifs seulement**

Ecrire un programme qui permet de saisir des nombres et qui calcule la somme et la moyenne des nombres positifs seulement jusqu'à ce que l'utilisateur tape la valeur 0.

**Exemple d'exécution :**

```
Saisir un nombre : 2  
Saisir un nombre : 3  
Saisir un nombre : -5  
Saisir un nombre : 7  
Saisir un nombre : -7  
Saisir un nombre : 0  
La somme des nombres positifs : 12.00  
La moyenne des nombres positifs : 4.00
```

### **5-8) Exercice 8 : Le nombre d'apparition d'un nombre positif**

Ecrire un programme qui permet de saisir des nombres jusqu'à ce que l'utilisateur saisisse 0. Ce programme calcule le nombre d'apparition des nombres positifs.

**Exemple d'exécution :**

```
Saisir un nombre : 2  
Saisir un nombre : 3  
Saisir un nombre : -5  
Saisir un nombre : 7
```

Saisir un nombre : 0  
3 nombres positifs

### **5-9) Exercice 9 : Le PGCD**

Ecrire un programme qui permet de saisir deux entiers positifs et qui calcule leur PGCD (Plus grand commun diviseur).

PGCD (a,b) = PGCD (b,r) tel que r=a%b

**Exemple d'exécution :**

Saisir un entier positif: 12  
Saisir un entier positif: 20  
PGCD (12,20) = 4

### **5-10) Exercice 10 : Le quotient et le reste sans utiliser / et %**

Ecrire un programme qui permet de saisir deux entiers positifs et qui calcule le quotient et le reste sans utiliser les opérateurs / et %.

**Exemple d'exécution :**

Saisir un entier positif: 20  
Saisir un entier positif: 7  
20 / 7 = 2  
20 mod 7 = 6

### **5-11) Exercice 11 : Nombres premiers entre eux**

Ecrire un programme qui permet de saisir deux entiers positifs et qui vérifie s'ils sont premiers entre eux ou non.

**Exemple d'exécution :**

Saisir un entier positif: 20  
Saisir un entier positif: 7  
20 et 7 sont premiers entre eux

### **5-12) Exercice 12 : Un nombre est en octal ou non**

Ecrire un programme qui permet de vérifier si un nombre est en octal.

**Exemple d'exécution :**

Saisir un nombre octal: 258  
Ce n'est pas un nombre octal

### **5-13) Exercice 13 : Un nombre est en binaire ou non**

Ecrire un programme qui permet de vérifier si un nombre est en binaire.

**Exemple d'exécution :**

Saisir un nombre binaire: 100011  
C'est un nombre binaire

#### **5-14) Exercice 14 : Conversion décimale - octale**

Ecrire un programme qui permet de saisir un entier en décimale et qui donne sa valeur en octale

**Exemple d'exécution :**

Saisir un nombre décimal: 258  
Le résultat en octal : 402

#### **5-15) Exercice 15 : Conversion décimale - binaire**

Ecrire un programme qui permet de saisir un entier en décimale et qui donne sa valeur en binaire

**Exemple d'exécution :**

Saisir un nombre décimal: 258  
Le résultat en binaire : 100000010

#### **5-16) Exercice 16 : Conversion binaire - décimale**

Ecrire un programme qui permet de saisir un entier en binaire et qui donne sa valeur en décimale

**Exemple d'exécution :**

Saisir un nombre binaire: 100000010  
Le résultat en décimal : 258

#### **5-17) Exercice 17 : Conversion octale - décimale**

Ecrire un programme qui permet de saisir un entier en octale et qui donne sa valeur en décimale

**Exemple d'exécution :**

Saisir un nombre octal: 402  
Le résultat en décimal : 258

## 6. La programmation modulaire : les fonctions

### 6-1) Exercice 1 : Fonction produit

1. Ecrire une fonction « **Produit** » qui permet de calculer le produit de deux réels.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche leur produit en utilisant la fonction « **Produit** ».

### 6-2) Exercice 2 : Fonction surface d'un cercle

1. Ecrire une fonction « **SurfaceCercle** » qui permet de calculer la surface d'un cercle.
2. Ecrire un programme qui demande à l'utilisateur de saisir le rayon d'un cercle et qui affiche sa surface en utilisant la fonction « **SurfaceCercle** ».

### 6-3) Exercice 3 : Fonction échange

1. Ecrire une fonction « **Echange** » qui permet d'échanger les valeurs de deux réels.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux nombres et qui les affiche après l'échange en utilisant la fonction « **Echange** ».

### 6-4) Exercice 4 : Fonction si pair ou non

1. Ecrire une fonction « **SiPair** » qui permet de vérifier si un nombre est pair ou impair. La fonction renvoie la valeur 1 si le nombre est pair et la valeur -1 s'il est impair.
2. Ecrire un programme qui demande à l'utilisateur de saisir un entier et qui affiche s'il est pair ou impair en utilisant la fonction « **SiPair** ».

### 6-5) Exercice 5 : Fonction signe

1. Ecrire une fonction « **Signe** » qui permet de vérifier si un nombre est positif, négatif ou nul. La fonction renvoie la valeur 1 si le nombre est positif, la valeur -1 s'il est négatif et la valeur 0 s'il est nul.
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre et qui affiche s'il est positif, négatif ou nul en utilisant la fonction « **Signe** ».

**6-6) Exercice 6 : fonction si multiple ou non**

1. Ecrire une fonction « **Multiple** » qui permet de vérifier si un entier est un multiple de « **n** » ou non. Si oui, la fonction renvoie la valeur 1. Sinon la fonction renvoie la valeur 0.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux entiers **a** et **n**. Ce programme affiche un message confirmant si **a** est un multiple de **n** ou non en utilisant la fonction « **Multiple** ».

**6-7) Exercice 7 : Fonction si diviseur ou non**

1. Ecrire une fonction « **Diviseur** » qui permet de vérifier si un entier est un diviseur de « **n** » ou non. La fonction renvoie la valeur 1 si oui. Sinon la fonction renvoie la valeur 0.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux entiers **a** et **n**. Ce programme affiche un message confirmant si **a** est un diviseur de **n** ou non en utilisant la fonction « **Diviseur** ».

**6-8) Exercice 8 : Fonction maximum**

1. Ecrire une fonction « **Max** » qui permet de renvoyer le maximum de deux nombres.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux nombres réelles et qui affiche le maximum en utilisant la fonction « **Max** ».
3. Ecrire un programme qui demande à l'utilisateur de saisir trois nombres réelles et qui affiche le maximum en utilisant la fonction « **Max** ».

**6-9) Exercice 9 : Fonction inverse**

1. Ecrire une fonction « **Inverse** » qui calcule l'inverse d'un nombre réel.
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre et qui affiche son inverse en utilisant la fonction « **Inverse** ».

**6-10) Exercice 10 : Fonction Message n fois**

1. Ecrire une fonction « **Message** » qui permet d'afficher le message « Bonjour Monsieur QARA » **n** fois.
2. Ecrire un programme qui permet d'afficher le message « Bonjour Monsieur QARA » 10 fois en utilisant la fonction « **Message** ».

**6-11) Exercice 11 : Fonction somme S1**

1. Ecrire une fonction « **Somme** » qui permet de calculer la somme  
$$S(n) = 1 + 2 + 3 + 4 + 5 + 6 + \dots + n$$

2. Ecrire un programme qui permet de calculer la somme S selon n saisi par l'utilisateur en utilisant la fonction « Somme ».

### 6-12) Exercice 12 : Fonction Factorielle

1. Ecrire une fonction « Fact » qui permet de calculer  $n!$
2. Ecrire un programme qui demande à l'utilisateur de saisir un entier positif n et qui affiche  $n!$  en utilisant la fonction « Fact ».

### 6-13) Exercice 13 : Fonction si premier ou non

1. Ecrire une fonction « SiPremier» qui permet de vérifier si un nombre est premier ou non. La fonction renvoie la valeur 1 s'il est premier. Sinon, elle renvoie la valeur 0.
2. Ecrire un programme qui demande à l'utilisateur de saisir un entier positif n et qui affiche s'il est premier ou non en utilisant la fonction « SiPremier ».

### 6-14) Exercice 14 : Fonction puissance

1. Ecrire une fonction « Puissance» qui permet de calculer la puissance  $x^y$ .
2. Ecrire un programme qui demande à l'utilisateur de saisir deux entiers a et n ( $n>0$ ) et qui affiche la puissance  $a^n$  en utilisant la fonction « Puissance ».

### 6-15) Exercice 15 : Fonction qui affiche une forme

1. Ecrire une fonction « Etoile» qui permet d'afficher le schéma (Le nombre de lignes est donné en paramètres)  
+  
++  
+++  
++++
2. Ecrire un programme qui demande à l'utilisateur de saisir le nombre de lignes et qui affiche le schéma ci-dessus selon le nombre saisi.

### 6-16) Exercice 16 : Fonction de contrôle de saisie

1. Ecrire une fonction « LirePositif» qui permet de saisir un nombre positif. Si le nombre saisi n'est pas positif la fonction demande la saisie encore.
2. Ecrire un programme qui demande à l'utilisateur de :
  - ✓ Saisir un nombre positif jusqu'à ce que la réponse convienne en utilisant la fonction « LirePositif ».
  - ✓ Afficher le nombre.

**6-17) Exercice 17 : fonction PGCD**

1. Ecrire une fonction « PGCD » qui permet de calculer le PGCD de deux entiers.
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre positif et qui affiche leur PGCD en utilisant la fonction « PGCD ».

**6-18) Exercice 18 : Fonction si octal ou non**

1. Ecrire une fonction « SiOctal » qui permet de vérifier si un nombre est en octal ou non. La fonction renvoie la valeur 1 si le nombre est en octal. Sinon, elle renvoie 0.
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre positif et qui affiche s'il est en octal ou non en utilisant la fonction « SiOctal ».

**6-19) Exercice 19 : Fonction si binaire ou non**

1. Ecrire une fonction « SiBinaire » qui permet de vérifier si un nombre est en binaire ou non. La fonction renvoie la valeur 1 si le nombre est en binaire. Sinon, elle renvoie 0.
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre positif et qui affiche s'il est en binaire ou non en utilisant la fonction « SiBinaire ».

**6-20) Exercice 20 : Fonction conversion décimale-octale**

1. Ecrire une fonction « DecOct » qui permet de convertir un nombre décimal en octal
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre décimal et qui affiche son équivalent octal en utilisant la fonction « DecOct ».

**6-21) Exercice 21 : Fonction conversion décimale-binaire**

1. Ecrire une fonction « DecBin » qui permet de convertir un nombre décimal en binaire
2. Ecrire un programme qui demande à l'utilisateur de saisir un nombre décimal et qui affiche son équivalent binaire en utilisant la fonction « DecBin »

**6-22) Exercice 22 : Récursivité**

1. Ecrire une fonction récursive qui permet de calculer le PGCD de deux entiers.
2. Ecrire un programme de test.

## 7. Les tableaux

### 7-1) Exercice 1 : Saisie et affichage d'un tableau

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 réels.
- Affiche le tableau.

### 7-2) Exercice 2 : Calcul de la valeur maximale d'un tableau

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 réels.
- Calcule la valeur maximale.
- Affiche la valeur maximale

### 7-3) Exercice 3 : Recherche dans un tableau

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 réels.
- Demande à l'utilisateur de saisir un nombre n.
- Cherche si le nombre existe dans le tableau ou non.

### 7-4) Exercice 4 : Affichage filtré des éléments d'un tableau

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 réels.
- Affiche les nombres positifs du tableau.

### 7-5) Exercice 5 : Le nombre d'apparition dans un tableau

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Affiche les nombres pairs
- Afficher le nombre d'apparition des nombres impairs.

### 7-6) Exercice 6\* : Conversion décimale - binaire

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un nombre entier positif.
- Calcule son équivalent binaire en utilisant les tableaux.

**7-7) Exercice 7 : Inverser les éléments d'un tableau**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Inverse le tableau et affiche le (Utiliser un nouveau tableau comme intermédiaire).
- Inverse le tableau et affiche le (N'utiliser pas un nouveau tableau comme intermédiaire)\*.

5	7	3	8	4
4	8	3	7	5

**7-8) Exercice 8 : Changer certaines valeurs d'un tableau**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Remplace les nombres négatifs par des 0.
- Affiche le nouveau tableau.

**7-9) Exercice 9 : La somme et la moyenne d'un tableau**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Calcule la somme des éléments du tableau et leur moyenne.
- Affiche le résultat

**7-10) Exercice 10 : Suppression des cases d'un tableau**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Supprime les nombres négatifs.
- Affiche le nouveau tableau et préciser sa dimension.

**7-11) Exercice 11 : Tri à bulle croissant**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Trie le tableau par ordre croissant (Tri à bulles).
- Affiche le nouveau tableau trié.

**7-12) Exercice 12 : Tri à bulle décroissant**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Trie le tableau par ordre décroissant (Tri à bulles).
- Affiche le nouveau tableau trié.

**7-13) Exercice 13 : Tri par sélection croissant**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Trie le tableau par ordre croissant (Tri par sélection).
- Affiche le nouveau tableau trié.

**7-14) Exercice 14 : Tri par sélection décroissant**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau de 10 entiers.
- Trie le tableau par ordre décroissant (Tri par sélection).
- Affiche le nouveau tableau trié.

**7-15) Exercice 15 : Manipulation des notes des stagiaires**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir les notes de 10 stagiaires.
- Trie le tableau par ordre de mérite (Tri à bulles).
- Calcule la moyenne de la classe.
- Affiche les notes et le nombre des stagiaires ayant une note supérieure à la moyenne de la classe.
- Affiche la note du majorant de la classe.

**7-16) Exercice 16 : Tableau à deux dimensions**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un tableau d'entiers de 5 lignes et 3 colonnes.
- Affiche le tableau
- Compte le nombre d'apparition des nombres positifs.
- Affiche le maximum.

**7-17)      Exercice 17 : La trace d'une matrice**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une matrice carrée d'entiers (Tableau à deux dimensions dont le nombre de lignes est égal au nombre de colonnes) de 4 lignes et 4 colonnes.
- Calcule la trace de la matrice (La trace est la somme de la diagonale principale de la matrice)
- Affiche la trace.

## 8. Les chaînes de caractères

### 8-1) Exercice 1: Saisie et affichage d'une chaine

Ecrire un programme qui :

- Demande à l'utilisateur de saisir son nom.
- Affiche le nom saisi et la longueur du nom.

### 8-2) Exercice 2 : Copie d'une chaine

Ecrire un programme qui :

- Demande à l'utilisateur de saisir son nom.
- Copie le nom dans une autre chaine de caractères « ch ».
- Affiche la chaine « ch ».

### 8-3) Exercice 3 : Affichage d'une partie d'une chaine

Ecrire un programme qui :

- Demande à l'utilisateur de saisir son nom et son prénom dans une seule chaine de caractères séparés par un espace (NOM PRENOM).

**Exemple :**

QARA Mohamed

- Affiche le prénom seulement (Dans notre exemple : MOHAMED)

### 8-4) Exercice 4 : Affichage par ordre alphabétique de deux chaines

Ecrire un programme qui :

- Demande à l'utilisateur de saisir deux chaines de caractères « ch1 » et « ch2 ».
- Affiche les chaines par ordre alphabétique.

### 8-5) Exercice 5 : Ignorance de la casse lors de la comparaison de deux chaines

Ecrire un programme qui :

- Demande à l'utilisateur de saisir deux chaines de caractères « ch1 » et « ch2 ».
- Affiche les chaines par ordre alphabétique en ignorant la casse (Sans tenir compte du minuscule et majuscule).

### **8-6) Exercice 6 : Concaténation de deux chaines**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir deux chaines de caractères « ch1 » et « ch2 ».

**Exemple :**

QARA

MOHAMED

- Affiche les chaines séparées par un espace (QARA MOHAMED).

### **8-7) Exercice 7 : Chaine à l'envers**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une chaine de caractères « ch1 ».

**Exemple :** QARA

- Affiche la chaine à l'envers (ARAQ).

### **8-8) Exercice 8 : Chaine palindrome**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une chaine de caractère ch.
- Affiche si la chaine est un palindrome ou non (Un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche).

### **8-9) Exercice 9 \*: Nombre de mot d'une chaine**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une chaine de caractère ch.
- Affiche le nombre de mots dans la chaine de caractères (un mot est séparé d'un autre par un espace ou plusieurs espaces).

### **8-10) Exercice 10 \*: Suppression d'un caractère de toute la chaine**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir un caractère « c » et une chaine de caractère « ch ».
- Elimine toutes les occurrences de « c » dans « ch ».
- Affiche la nouvelle chaine.

**8-11) Exercice 11 : conversion minuscule-majuscule**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une chaîne de caractères « ch ».
- Convertit toutes les majuscules dans des minuscules et vice versa.  
(QARA Mohamed → qara mOHAMED)
- Affiche la nouvelle chaîne.

**8-12) Exercice 12 : Remplacer une chaîne par une autre dans une chaîne**

Ecrire un programme qui :

- Demande à l'utilisateur de saisir une chaîne de caractères « ch » et 2 caractères c1 et c2.
- Remplace dans la chaîne « ch » toutes les occurrences de « c1 » par « c2 ».
- Affiche la nouvelle chaîne.

**8-13) Exercice 13 : Chiffrement et déchiffrement d'une chaîne**

- Ecrire une fonction « encrypt » qui permet de décaler toutes les lettres d'une chaîne de caractère d'un certain nombre de caractères.
- Ecrire une fonction « decrypt » qui permet de faire l'opération inverse.
- Ecrire un programme qui demande à l'utilisateur de
  - ✓ saisir une chaîne.
  - ✓ Saisir la clé de chiffrement (Le nombre de décalage)
  - ✓ Affiche le résultat (La chaîne chiffrée)
  - ✓ Fait l'opération inverse (le déchiffrement) et affiche le résultat.

## 9. Les structures

### 9-1) Exercice 1 : La manipulation d'un nombre complexe

- 1) Définir une structure « **complexe** ».
- 2) Définir une fonction « **afficher** » qui permet d'afficher un complexe.
- 3) Définir une fonction « **lire** » qui permet de saisir un complexe.
- 4) Définir une fonction « **somme** » pour calculer la somme de deux complexes.
- 5) Définir une fonction « **produit** » qui permet de calculer le produit de deux complexes.
- 6) Définir une fonction « **conjugue** » qui permet de calculer le conjugué d'un complexe.
- 7) Définir une fonction « **module** » qui permet de calculer le module d'un complexe.
- 8) Définir une fonction « **inverse** » qui permet de calculer l'inverse d'un complexe.
- 9) Définir une fonction « **div** » qui permet de calculer le résultat de la division de deux complexes.
- 10) Ecrire un programme qui demande à l'utilisateur de saisir deux complexes  $z_1$  et  $z_2$  et qui calcule et affiche  $z_1+z_2$ ,  $z_1 \times z_2$  et  $z_1/z_2$  en utilisant les fonctions précédemment définies.

### 9-2) Exercice 2 : La manipulation d'un nombre rationnel

- 1) Définir une structure « **rationnel** »
- 2) Ecrire une fonction « **somme** » qui permet de calculer la somme de deux rationnels.
- 3) Ecrire une fonction « **produit** » qui permet de calculer le produit de deux rationnels
- 4) Ecrire une fonction « **PGCD** » qui permet de calculer le plus grand commun diviseur de deux entiers.
- 5) Ecrire une fonction « **Simplifier** » qui permet de simplifier l'écriture d'un rationnel.
- 6) Ecrire un programme qui demande à l'utilisateur de saisir deux rationnels et qui affiche leur produit et leur somme d'une manière simplifiée.

### **9-3) Exercice 3 : La manipulation d'une structure stagiaire**

- 1) Définir une structure « stagiaire » qui définit un stagiaire (nom, prénom, note et date de naissance).
- 2) Ecrire une fonction « afficher » qui permet d'afficher les informations d'un stagiaire.
- 3) Ecrire une fonction « lire » qui permet de saisir les informations d'un stagiaire.
- 4) Ecrire une fonction « max » qui permet de retourner le stagiaire ayant la meilleure note.
- 5) Ecrire une fonction « moyenne » qui permet de calculer la moyenne des notes d'un tableau de stagiaires.
- 6) Ecrire une fonction « trimerite » qui permet de trier par ordre de mérite un tableau de stagiaires.
- 7) Ecrire une fonction « trialpha » qui permet de trier par ordre alphabétique un tableau de stagiaires.
- 8) Ecrire un programme qui demande à l'utilisateur de saisir un tableau de stagiaires et qui affiche :
  - ✓ La moyenne des notes.
  - ✓ Le stagiaire ayant la meilleure note.
  - ✓ Les stagiaires ayant une note supérieure ou égale à cette moyenne par ordre de mérite.

### **9-4) Exercice 4 : La manipulation d'une structure « heure »**

- 1) Définir une structure « heure » (heure et minute).
- 2) Ecrire une fonction « afficher » qui permet d'afficher une heure.
- 3) Ecrire une fonction « lire » qui permet de saisir une heure.
- 4) Ecrire une fonction « siheure » qui permet de vérifier la validité d'une heure.
- 5) Ecrire une fonction « plus1 » qui permet d'ajouter une minute.
- 6) Ecrire une fonction « moins1 » qui permet de soustraire une minute.
- 7) Ecrire une fonction « duree » qui permet de calculer la durée entre deux heures.

8) Ecrire un programme qui demande à l'utilisateur de saisir une heure de départ et une heure d'arrivée et qui calcule la durée de voyage (contrôler la saisie des heures).

### **9-5) Exercice 5 : La manipulation d'une structure « date »**

- 1) Définir une structure « date ».
- 2) Ecrire une fonction « afficher » qui permet d'afficher une date.
- 3) Ecrire une fonction « lire » qui permet de saisir une date.
- 4) Ecrire une fonction « sibissextile » qui permet de vérifier si une année est bissextile ou non.
- 5) Ecrire une fonction « plus1 » qui permet d'ajouter jour.
- 6) Ecrire une fonction « moins1 » qui permet de soustraire un jour.
- 7) Ecrire une fonction « duree » qui permet de calculer la durée entre deux dates.
- 8) Ecrire un programme qui demande à l'utilisateur de saisir sa date de naissance et la date actuelle et qui calcule son âge (nombre de jours).

### **9-6) Exercice 6 : La manipulation d'une adresse IPv4**

- 1) Définir une structure « ip » qui définit une adresse IP v4.
- 2) Ecrire une fonction « showip » qui permet d'afficher une adresse IP.
- 3) Ecrire une fonction « readip » qui permet de saisir une adresse IP.
- 4) Ecrire une fonction « ifip » qui permet de vérifier la validité d'une adresse IP.
- 5) Ecrire une fonction « ifmask » qui permet de vérifier la validité d'un masque (à barre oblique).
- 6) Ecrire une fonction « classe » qui permet de retourner la classe d'une adresse IP.
- 7) Ecrire une fonction « maskdec » qui permet de convertir un masque à barre oblique en un masque décimal.
- 8) Ecrire une fonction « netip » qui permet de calculer l'adresse réseau.
- 9) Ecrire une fonction « broadip » qui permet de calculer l'adresse de diffusion.

- 
- 10) Ecrire une fonction « firstip » qui permet de calculer la première adresse.
  - 11) Ecrire une fonction « lastip » qui permet de calculer la dernière adresse.
  - 12) Ecrire un programme qui demande à l'utilisateur de saisir une adresse IP et un masque à barre oblique et qui calcule l'adresse réseau, l'adresse de diffusion, la première adresse et la dernière adresse.

## 10. Les fichiers

### 10-1) Exercice 1 : Lecture et écriture dans un fichier

1. Ecrire un programme qui demande à l'utilisateur de saisir son nom et le sauvegarder dans un fichier nom.txt.
2. Afficher le contenu de ce fichier.

### 10-2) Exercice 2 : Ecriture avec sauvegarde du contenu

1. Ecrire un programme qui permet de sauvegarder des noms séparés par des retours à la ligne dans un fichier noms.txt chaque fois qu'il est ouvert.
2. Afficher le contenu de ce fichier.

### 10-3) Exercice 3 : Ecriture en boucle

Ecrire un programme qui permet de sauvegarder plusieurs noms séparés par des retours à la ligne dans un fichier noms.txt. La saisie s'arrête lorsque l'utilisateur entre le caractère « \* » qui ne sera pas sauvegardé.

### 10-4) Exercice 4 : La taille d'un fichier

1. Ecrire une fonction qui permet de calculer la taille dans un fichier.
2. Ecrire un programme qui demande à l'utilisateur de saisir le nom d'un fichier et qui affiche sa taille en octets.

### 10-5) Exercice 5 : Recherche d'un caractère et son nombre d'apparition

1. Ecrire une fonction qui permet de chercher un caractère dans le fichier noms.txt. La fonction retourne le nombre de caractères trouvés.
2. Ecrire un programme qui demande à l'utilisateur de saisir un caractère et qui affiche son nombre d'apparitions dans le fichier noms.txt.

### 10-6) Exercice 6 : Recherche d'une chaîne de caractères et son nombre d'apparition

1. Ecrire une fonction qui permet de chercher une chaîne de caractères dans le fichier noms.txt. La fonction retourne le nombre de chaînes trouvées.
2. Ecrire un programme qui demande à l'utilisateur de saisir un caractère et qui affiche son nombre d'apparitions dans le fichier noms.txt.

**10-7) Exercice 7 : Nombre de lignes d'un fichier**

1. Ecrire une fonction qui calcule le nombre de lignes dans le fichier noms.txt.
2. Ecrire un programme qui demande à l'utilisateur de saisir le nom d'un fichier et qui affiche tous le nombre de lignes de ce fichier.

**10-8) Exercice 8 : Affichage filtré d'une partie du fichier**

1. Ecrire une fonction qui permet d'afficher les noms qui commencent par un caractère dans le fichier noms.txt. Ce caractère est saisi par l'utilisateur.
2. Ecrire un programme qui demande à l'utilisateur de saisir un caractère et qui affiche tous les noms qui commencent par ce caractère.

**10-9) Exercice 9 : Remplacer un caractère dans un fichier**

1. Ecrire une fonction qui permet de remplacer un caractère par un autre dans un fichier.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux caractères « **c1** » et « **c2** » et qui remplace toutes les occurrences de « **c1** » dans un fichier par « **c2** ».

**10-10) Exercice 10 : Remplacer une chaîne dans un fichier**

1. Ecrire une fonction qui permet de remplacer une chaîne de caractères par une autre dans un fichier.
2. Ecrire un programme qui demande à l'utilisateur de saisir deux chaînes de caractères « **ch1** » et « **ch2** » et qui remplace toutes les occurrences de « **ch1** » dans un fichier par « **ch2** ».

**10-11) Exercice 11 : Supprimer un caractère d'un fichier**

1. Ecrire une fonction qui permet de supprimer toutes les occurrences d'un caractère dans un fichier.
2. Ecrire un programme qui demande à l'utilisateur de saisir un caractère et qui supprime toutes les occurrences de ce caractère dans un fichier (Afficher le contenu du fichier avant et après la suppression)

**10-12) Exercice 12 : Supprimer une chaîne d'un fichier**

1. Ecrire une fonction qui permet de supprimer toutes les occurrences d'une chaîne de caractère dans un fichier.
2. Ecrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et qui supprime toutes les occurrences de cette chaîne dans un fichier (Afficher le contenu du fichier avant et après la suppression).

**Corrigés des exercices**



**CORRIGÉS DES  
EXERCICES**

## 1. Les variables et les opérateurs :

### 1-1) Exercice 1-1 :

	a	b	c	d
<b>int a=15, b=3, c, d;</b>	15	3	-	-
<b>a=a-b;</b>	12	3	-	-
<b>b=a/b;</b>	12	4	-	-
<b>c=a+b*4; (1)</b>	12	4	28	-
<b>d=c%4;</b>	12	4	28	0

(1) On calcule le produit  $b*4$  et après on calcule la somme

### 1-2) Exercice 2-1 :

	a	b	c	d
<b>int a=15, b=3, c, d;</b>	15	3	-	-
<b>a=(a&gt;=b); (1)</b>	1	3	-	-
<b>b=(a==b); (2)</b>	1	0	-	-
<b>c=(a!=b); (3)</b>	1	0	1	-
<b>d=(a=5); (4)</b>	5	0	1	5

- (1)  $(a>=b)$  est vraie. Donc le résultat de la comparaison est 1.
- (2)  $(a==b)$  est fausse. Donc le résultat de la comparaison est 0.
- (3)  $(a!=b)$  est vraie. Donc le résultat de la comparaison est 1.
- (4)  $(a=b)$  vaut 5.

### 1-3) Exercice 3-1 :

	a	b	c	d
<b>int a=15, b=3, c, d;</b>	15	3	-	-
<b>a= (b!=a) &amp;&amp; (b&gt;2); (1)</b>	1	3	-	-
<b>b= (b=4)    (a&gt;5); (2)</b>	1	1	-	-
<b>c= !(a&lt;=15); (3)</b>	1	1	0	-
<b>d= !(c==0) &amp;&amp; (b!=a); (4)</b>	1	1	0	0

- (1)  $(b!=a)$  est vraie.  $(b>2)$  est vraie. Vraie et Vraie  $\rightarrow$  Vraie  $\rightarrow$  1.
- (2)  $(b=4)$  est vraie.  $(a>5)$  est vraie. Vraie ou Vraie  $\rightarrow$  Vraie  $\rightarrow$  1.
- (3)  $(a<=15)$  est vraie. Donc  $!(a<=15)$  est fausse  $\rightarrow$  0.
- (4)  $(c==0)$  est vraie  $\rightarrow$   $!(c==0)$  est fausse.  $(b!=a)$  est fausse  $\rightarrow$  0

### 1-4) Exercice 4-1 :

	a	b	c	d
<b>int a=15, b=3, c, d;</b>	15	3	-	-
<b>a= a &amp; 10;</b>	10	3	-	-
<b>b= 14   b;</b>	10	15	-	-
<b>c= a &amp; b;</b>	10	15	10	
<b>d= c ^ b;</b>	10	15	10	5

Explication :

- **a = a & 10**

✓ a =  $(15)_{10} = (1111)_2$

✓  $(10)_{10} = (1010)_2$

a	1	1	1	1	15
10	1	0	1	0	10
a & 10	1	0	1	0	10

- **b = 14 | b**

✓ b =  $(3)_{10} = (0111)_2$

✓  $(14)_{10} = (1110)_2$

b	0	0	1	1	3
14	1	1	1	0	14
14   b	1	1	1	1	15

- **c = a & b**

✓ a =  $(10)_{10} = (1010)_2$

✓ b =  $(15)_{10} = (1111)_2$

a	1	0	1	0	10
b	1	1	1	1	15
a & b	1	0	1	0	10

- **d = c ^ b**

✓ c =  $(10)_{10} = (1010)_2$

✓ b =  $(15)_{10} = (1111)_2$

c	1	0	1	0	10
b	1	1	1	1	15
c ^ b	0	1	0	1	5

### 1-5) Exercice 5-1 :

	a	b	c	d	e	f
<b>int a=20, b=3,c;</b>	20	3	-	-	-	-
<b>c=a/b; (1)</b>	20	3	6	-	-	-
<b>d=a/b; (2)</b>	20	3	6	6.0	-	-
<b>e=(float)a/b; (3)</b>	20	3	6	6.0	6.67	-
<b>f=(float)(a/b); (4)</b>	20	3	6	6.0	6.67	6.0

- (1)  $20 / 3 = 6$
- (2)  $20 / 6 = 6.0$
- (3)  $20 / 6 = 6.67$
- (4) La priorité est pour / d'où le résultat est 6.0

### 1-6) Exercice 6-1 :

	a	b	c	d	e
<b>int a=10,b=7,c,d,e;</b>	10	7	-	-	-
<b>c=a/2*b; (1)</b>	10	7	35	-	-
<b>d=b/3*c; (2)</b>	10	7	35	0	-
<b>e=c+d/2; (3)</b>	10	7	35	0	35

- (1) On commence par / puis \*
- (2) (b/3)/c
- (3) On commence par / puis +

### 1-7) Exercice 7-1 :

```
#include <stdio.h>
main (){
    int a=5,b=3;
    // Variable intermédiaire
    int t;
    // L'échange
    t=a;
    a=b;
    b=t; }
```

**1-8) Exercice 8-1 :**

```
#include <stdio.h>
main (){
    int a=5,b=3,c=7;
    // Variable intermédiaire
    int t;
    // La permutation
    t=a;
    a=c;
    c=b;
    b=t;
    printf ("%d %d %d",a,b,c);
}
```

## 2. Les instructions d'entrées sorties conversationnelles

### 2-1) Exercice 1-2 :

```
#include <stdio.h>
#include <conio.h>

main ()
{
    printf ("BonjourMonsieur\nMerci");
    getch();
}
```

### 2-2) Exercice 2-2 :

```
#include <stdio.h>
#include <conio.h>

main ()
{
    printf (" 5\t 10\t150\t 7\n120\t 1\t 14\t 78");
    getch();
}
```

### 2-3) Exercice 3-2 :

```
#include <stdio.h>
#include <conio.h>

main ()
{
    //Déclaration des variables
    int a,b,s;
    //Saisie de deux entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    //Calcul de la somme
    s=a+b;
    // Affichage de la somme
    printf ("La somme est : %d",s);
    getch();
}
```

**2-4) Exercice 4-2 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    //Déclaration des variables
    int a,b,p;
    //Saisie de deux entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    //Calcul du produit
    p=a*b;
    // Affichage du produit
    printf ("Le produit est : %d",p);
    getch();
}
```

**2-5) Exercice 5-2 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    //Déclaration des variables
    int a,b,q,r;
    //Saisie de deux entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    //Calcul du quotient
    q=a/b;
    //Calcul du reste
    r=a%b;
    // Affichage du quotient et du reste
    printf ("Le quotient est : %d\nLe reste est : %d",q,r);
    getch();
}
```

**2-6) Exercice 6-2 :**

```
#include <stdio.h>
#include <conio.h>

main (){
    //Déclaration des variables
    int a,b;
    float x;
    //Saisie de deux entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    //Calcul du résultat de la division exacte
    x=(float)a/b;
    // Affichage du résultat
    printf ("%d / %d = %.2f",a,b,x);
    getch();
}
```

**2-7) Exercice 7-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    //Déclaration des variables
    int a,b;
    //Déclaration d'une variable temporaire
    int t;
    //Saisie de deux entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    //Affichage avant l'échange
    printf ("Avant:\na = %d\tb = %d\n",a,b);
    //L'échange
    t=a;
    a=b;
    b=t;
    //Affichage après l'échange
    printf ("Après:\na = %d\tb = %d",a,b);
    getch();
}
```

**2-8) Exercice b-z :**

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration des variables
    char a,b,c;
    printf ("Saisir un caractere a : ");
    scanf("%c",&a);
    //Vidage du Buffer
    fflush(stdin);
    printf ("Saisir un caractere b : ");
    scanf("%c",&b);
    fflush(stdin);
    //Saisie du 3ème caractère
    printf ("Saisir un caractere c : ");
    scanf("%c",&c);
    //Affichage des 3 caractères
    printf ("a = %c\nb = %c\nnc = %c",a,b,c);
    getch();
}

#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration des variables
    char a,b,c;
    //Saisie du 1er caractère
    printf ("Saisir un caractere a : ");
    a = getchar();
    //Vidage du Buffer
    fflush(stdin);
    //Saisie du 2ème caractère
    printf ("Saisir un caractere b : ");
    b = getchar ();
    //Vidage du Buffer
    fflush(stdin);
    //Saisie du 3ème caractère
    printf ("Saisir un caractere c : ");
    c = getchar();
    //Affichage des 3 caractères
    printf ("a = ");
    putchar(a);
    printf ("\nb = ");
    putchar(b);
    printf ("\nnc = ");
    putchar(c);
    getch();
}
```

**2-9) Exercice 5-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration des variables
    int a,b,c;
    //Déclaration d'une variable temporaire
    int t;
    //Saisie de 3 entiers
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    printf ("Saisir un entier b:");
    scanf("%d",&b);
    printf ("Saisir un entier c:");
    scanf("%d",&c);
    //Affichage avant la permutation
    printf ("Avant:\n a = %d\n b = %d\n c = %d\n",a,b,c);
    //La permutation
    t=a;
    a=b;
    b=c;
    c=t;
    //Affichage après la permutation
    printf ("Après:\n a = %d\n b = %d\n c = %d",a,b,c);
    getch(); }
```

**2-10) Exercice 10-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    printf ("Taille d'un entier : %d octet(s)\n",sizeof(int));
    printf ("Taille d'un réel : %d octet(s)\n",sizeof(float));
    printf ("Taille d'un caractère : %d octet(s)\n",sizeof(char));
    getch();
}
```

**2-11) Exercice 11-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    char ch;
    printf ("Saisir un caractère : ");
    ch = getchar ();
    printf ("Le code ASCII de %c est : %d",ch,ch);
    getch(); }
```

**2-12) Exercice 12-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int h,m,s;
    int d;
    printf ("Saisir une duree (HH:MM:SS) : ");
    scanf ("%d:%d:%d",&h,&m,&s);
    d = 3600 * h + 60 * m + s;
    printf ("La durée en secondes est : %d secondes",d);
    getch();
}
```

**2-13) Exercice 13-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int h,m,s;
    int d;
    printf ("Saisir la durée en secondes : ");
    scanf ("%d",&d);
    h = d / 3600 ;
    m = ( d % 3600 ) / 60 ;
    s = ( d % 3600 ) % 60 ;
    printf ("La durée en (HH :MM :SS) est : %d : %d : %d",h,m,s);
    getch();
}
```

**2-14) Exercice 14-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int a,u;
    printf ("Saisir un entier a:");
    scanf("%d",&a);
    u = a % 10 ;
    printf ("Son unite est : %d", u);
    getch();
}
```

**2-15) Exercice 15-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    char min,maj;
    printf ("Saisir un caractere en minuscule :");
    scanf("%c",&min);
    maj = min - 32 ;
    printf ("Le caractere en majuscule est : %c", maj);
    getch();
}
```

**2-16) Exercice 16-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float r;
    float s;
    printf ("Saisir le rayon d'un cercle :");
    scanf("%f",&r);
    s = 3.14 * r * r ;
    printf ("La surface du cercle est : %.2f", s);
    getch();
}
```

**2-17) Exercice 17-2 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float r;
    float s,v;
    printf ("Saisir le rayon d'une sphere :");
    scanf("%f",&r);
    s = 4 * 3.14 * r * r ;
    v = (4/3) * 3.14 * r * r * r ;
    printf ("La surface de la sphere est : %.2f\n", s);
    printf ("Le volume de la sphere est : %.2f\n", v);
    getch();
}
```

**2-18) Exercice 18-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    float r,h;
    float s,v;
    printf ("Saisir le rayon d'un cylindre :");
    scanf ("%f",&r);
    printf ("Saisir la hauteur d'un cylindre :");
    scanf ("%f",&h);
    s = 2 * 3.14 * r * (h + r) ;
    v = 3.14 * r * r * h ;
    printf ("La surface du cylindre est : %.2f\n", s);
    printf ("Le volume du cylindre : %.2f\n", v);
    getch();
}
```

**2-19) Exercice 19-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    float a,b,c;
    float moy;
    printf ("Saisir un nombre a : ");
    scanf ("%f",&a);
    printf ("Saisir un nombre b : ");
    scanf ("%f",&b);
    printf ("Saisir un nombre c : ");
    scanf ("%f",&c);
    moy = (a + b + c)/3;
    printf ("La moyenne est : %.2f\n", moy);
    getch();
}
```

**2-20) Exercice 20-2 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    float PrixTTC;
    float PrixHT;
    printf ("Saisir le prix hors taxe d'un article : ");
    scanf ("%f",&PrixHT);
    PrixTTC = 1.2 * PrixHT;
    printf ("Le prix TTC : %.2f\n", PrixTTC);
    getch();
}
```

### 3. Les instructions de contrôle - if et switch

#### 3-1) Exercice 1-3 :

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration d'un nombre entier
    int a;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    // Tester si le nombre est pair ou impair
    if(a%2==0) printf ("%d est pair",a);
    else printf ("%d est impair",a);
    getch();
}
```

#### 3-2) Exercice 2-3 :

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration d'un nombre
    int a;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    // Tester si le nombre est positif, négatif ou nul
    if(a>0) printf ("% .2f est positif",a);
    else if (a<0)printf ("% .2f est negatif",a);
    else printf ("% .2f est nul",a);
    getch();
}
```

#### 3-3) Exercice 3-3 :

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration d'un nombre entier
    int a;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    // Tester s'il est un multiple de 7 ou non
    if(a%7==0) printf ("%d est un multiple de 7",a);
    else printf ("%d n'est pas un multiple de 7",a);
    getch();
}
```

**3-4) Exercice 4-3 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration d'un nombre entier
    int a;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    // Tester si c'est un diviseur de 100 ou non
    if(100%a==0) printf ("%d est un diviseur de 100",a);
    else printf ("%d n'est pas un diviseur de 100",a);
    getch();
}
```

**3-5) Exercice 5-3 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    //Déclaration des variables
    float a,b;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    printf ("Saisir un nombre :");
    scanf ("%f",&b);
    // Comparer les deux nombres pour afficher le maximum
    if(a>b) printf ("Le maximum est : %.2f",a);
    else printf ("Le maximum est : %.2f",b);
    getch();
}
```

**3-6) Exercice 6-3 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    //Déclaration des variables
    float a,b,c;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    printf ("Saisir un nombre :");
    scanf ("%f",&b);
    printf ("Saisir un nombre :");
    scanf ("%f",&c);
    // Comparer les 3 nombres pour afficher le minimum
```

```

if(a<=b && a<=c) printf ("Le minimum est : %.2f",a);
else if (b<=a && b<=c) printf ("Le minimum est : %.2f",b);
else if (c<=a && c<=b) printf ("Le minimum est : %.2f",c);
getch();
}

```

### 3-7) Exercice 7-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    //Déclaration du nombre
    float a;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    // Tester pour afficher la valeur absolue
    if(a>=0) printf ("La valeur absolue de %.2f est : %.2f",a,a);
    else printf ("La valeur absolue de %.2f est : %.2f",a,-a);
    getch();
}

```

### 3-8) Exercice 8-3 :

```

#include <stdio.h>
#include <conio.h>

main (){
    float a,b;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    printf ("Saisir un nombre :");
    scanf ("%f",&b);
    if(a>0 && b>0 || a<0 && b<0) printf ("Le produit est positif");
    else if(a>0 && b<0 || a<0 && b>0) printf ("Le produit est négatif");
    else printf ("Le produit est nul");
    getch();
}

```

### 3-9) Exercice 9-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    float a;
    printf ("Saisir la note :");
    scanf ("%f",&a);
    if(a<10) printf ("Echec");
}

```

```

else if(a>=10 && a<12) printf ("La mention est passable");
else if(a>=12 && a<14) printf ("La mention est assez bien");
else if(a>=14 && a<16) printf ("La mention est bien");
else if(a>=16 && a<18) printf ("La mention est tres bien");
else if(a>=18 && a<=20) printf ("La mention est excellent");
getch();
}

```

### 3-10) Exercice 10-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    float t;
    printf ("Saisir la temperature :");
    scanf ("%f",&t);
    if(t<=0) printf ("L'eau est en etat solide");
    else if(t>0 && t<100) printf ("L'eau est en etat liquide");
    else printf ("L'eau est en etat gazeux");
    getch();
}

```

### 3-11) Exercice 11-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    float a;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    if(a!=0) printf ("L'inverse est : %.2f",1/a);
    else printf ("Erreur de division par 0");
    getch();
}

```

### 3-12) Exercice 12-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    float a,b,c;
    printf ("Saisir les longueurs des côtes du triangle :\n");
    printf ("a = ");
    scanf ("%f",&a);
    printf ("b = ");
    scanf ("%f",&b);
    printf ("c = ");
    scanf ("%f",&c);
}

```

```

printf ("Le triangle est .\n"),
//Les tests des côtés du triangle
if(a==b && a==c) printf ("equilateral\n");
else if(a==b || a==c || b==c) printf ("isocele\n");
else if (a*a==b*b+c*c || b*b==a*a+c*c || c*c==b*b+a*a)printf ("rectangle\n");
else printf ("normal");
getch();
}

```

### 3-13) Exercice 13-3 :

```

#include <stdio.h>
#include <conio.h>

main ()
{
    int h1,m1;
    // Nouvelle heure
    int h2,m2;
    // Variable pour la vérifier la validité de l'heure
    int x=0;
    printf ("Saisir une heure (HH:MM) : ");
    scanf ("%d : %d",&h1,&m1);

    if(m1<=58 && h1>=0 && h1<=23)
    {
        m2 = m1+1;
        h2 = h1;
        x=1;
    }
    else if(m1==59 && h1>=0 && h1<23)
    {
        m2 = 0;
        h2 = h1+1;
        x=1;
    }
    else if(m1==59 && h1==23){
        m2 = 0;
        h2 = 0;
        x=1;
    }
    //Si l'heure est valide afficher la sinon afficher "heure invalide"
    if(x==1) printf ("Nouvelle heure : %d : %d",h2,m2);
    else printf ("Heure invalide");
    getch();
}

```

## 3-14) Exercice 14-3 :

```
#include <stdio.h>
#include <conio.h>

main (){
    int h1,m1,s1;
    // Nouvelle heure
    int h2,m2,s2;
    // Variable pour la vérifier la validité de l'heure
    int x=0;
    printf ("Saisir une heure (HH:MM:SS) : ");
    scanf ("%d : %d : %d",&h1,&m1,&s1);
    if(s1<=58 && m1>=0 && m1<=59 && h1>=0 && h1<=23)
    {
        s2=s1+1;
        m2 = m1;
        h2 = h1;
        x=1;
    }
    else if(s1==59 && m1>=0 && m1<59 && h1>=0 && h1<=23)
    {
        s2 = 0;
        m2 = m1+1;
        h2 = h1;
        x=1;
    }
    else if(s1==59 && m1==59 && h1>=0 && h1<23)
    {
        s2 = 0;
        m2 = 0;
        h2 = h1+1;
        x=1;
    }
    else if(s1==59 && m1==59 && h1==23)
    {
        s2 = 0;
        m2 = 0;
        h2 = 0;
        x=1;
    }
    //Si l'heure est valide afficher la sinon afficher "heure invalide"

    if(x==1) printf ("Nouvelle heure : %d : %d : %d",h2,m2,s2);
    else printf ("Heure invalide");
    getch();
}
```

## 3-15) Exercice 15-3 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int h1,m1;
    // Nouvelle heure
    int h2,m2;
    // Variable pour la vérifier la validité de l'heure
    int x=0;
    printf ("Saisir une heure (HH:MM) : ");
    scanf ("%d : %d",&h1,&m1);
    if(m1>0 && h1>=0 && h1<=23) {
        m2 = m1-1;
        h2 = h1;
        x=1;
    }
    else if(m1==0 && h1>0 && h1<=23) {
        m2 = 59;
        h2 = h1-1;
        x=1;
    }
    else if(m1==0 && h1==0) {
        m2 = 59;
        h2 = 23;
        x=1;
    }
    //Si l'heure est valide afficher la sinon afficher "heure invalide"
    if(x==1) printf ("Nouvelle heure : %d : %d",h2,m2);
    else printf ("Heure invalide");
    getch();
}
```

## 3-16) Exercice 16-3 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int h1,m1,s1;
    // Nouvelle heure
    int h2,m2,s2;
    // Variable pour vérifier la validité de l'heure
    int x=0;
    printf ("Saisir une heure (HH:MM:SS) : ");
    scanf ("%d : %d : %d",&h1,&m1,&s1);
    if(s1>0 && s1<=59 && m1>=0 && m1<=59 && h1>=0 && h1<=23) {
        s2=s1-1;
        m2 = m1;
        h2 = h1;
        x=1;
    }
}
```

```

else if(s1==0 && m1>0 && m1<=23 && h1>0 && h1<=23) {
    s2 = 59;
    m2 = m1-1;
    h2 = h1;
    x=1;
}
else if(s1==0 && m1==0 && h1>0 && h1<=23) {
    s2 = 59;
    m2 = 59;
    h2 = h1-1;
    x=1;
}
else if(s1==0 && m1==0 && h1==0) {
    s2 = 59;
    m2 = 59;
    h2 = 23;
    x=1;
}
//Si l'heure est valide afficher la sinon afficher "heure invalide"
if(x==1) printf ("Nouvelle heure : %d : %d : %d",h2,m2,s2);
else printf ("Heure invalide");
getch();
}

```

### 3-17) Exercice 17-3 :

```

#include <stdio.h>
#include <conio.h>
main ()
{
    // Heure de départ
    int h1,m1;
    // Heure d'arrivée
    int h2,m2,s2;
    // Durée
    int hd,md;
    //Saisir l'heure de départ et l'heure d'arrivée
    printf ("Saisir l\\'heure de depart (HH:MM) : ");
    scanf ("%d : %d",&h1,&m1);
    printf ("Saisir l\\'heure d\\'arrivee (HH:MM) : ");
    scanf ("%d : %d",&h2,&m2);
    // Calcul de la durée
    if(m2<m1)
    {
        m2=m2+60;
        h2=h2-1;
    }
}

```

```

if(h2>=h1) {
    md=m2-m1;
    hd=h2-h1;
    printf ("%d : %d",hd,md);
}
else printf ("Invalide");
getch();
}

```

### 3-18) Exercice 18-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    // Heure de départ
    int h1,m1;
    // Heure d'arrivée
    int h2,m2,s2;
    // Durée de voyage (doit être <=24 heures)
    int hd,md;
    //Saisir l'heure de départ et la durée de voyage
    printf ("Saisir l\\'heure de départ (HH:MM) : ");
    scanf ("%d : %d",&h1,&m1);
    printf ("Saisir la duree de voyage (Heure:Minute) : ");
    scanf ("%d : %d",&hd,&md);
    // Calcul de l'heure d'arrivée
    m2=m1+md;
    if(m2>=60) {
        m2=m2-60;
        h2=h1+hd+1;
    }
    else h2=h1+hd;
    if(h2>=24) {
        h2=h2-24;
    }
    printf ("%d : %d",h2,m2);
    getch();
}

```

### 3-19) Exercice 19-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int a;
    //Saisir une année
    printf ("saisir une annee : ");
    scanf ("%d",&a);
}

```

```
// Tester si cette année est bissextile ou non
if(a%400==0 || a%4==0 && a%100!=0)
printf ("Cette année est bissextile");
else printf ("Cette année est non bissextile");
getch();
```

### 3-20) Exercice 20-3 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int valide = 1;
    int j,m,a;
    //Saisir une date
    printf ("Saisir une date (JJ/MM/AA) :");
    scanf ("%d/%d/%d",&j,&m,&a);
    // Calcul de la nouvelle date
    //Les mois de 30 jours
    if(m==4 || m==6 || m==9 || m==11)
    {
        if(j>=1 && j<30) j=j+1;
        else if (j==30) {
            j=1;
            m=m+1; }
        else valide = 0;
    }
    //Les mois de 31 jours sauf le mois 12
    else if (m==1 || m==3 || m==5 || m==7 || m==8 || m==10)
    {
        if(j>=1 && j<31) j=j+1;
        else if(j==31) {
            j=1;
            m=m+1;
        }
        else valide = 0;
    }
    //Le mois 12
    else if (m==12)
    {
        if(j>=1 && j<31) j=j+1;
        else if(j==31) {
            j=m=1;
            a=a+1;
        }
        else valide = 0;
    }
}
```

```

//Le mois 2
else if(m==2){
    //Année bissextile
    if(a%400==0 || a%4==0 && a%100!=0){
        if(j>=1 && j<29) j=j+1;
        else if (j==29) {
            j=1;
            m=m+1;
        }
        else valide = 0;
    }
    //Année non bissextile
    else {
        if(j>=1 && j<28) j=j+1;
        else if (j==28) {
            j=1;
            m=m+1;
        }
        else valide = 0;
    }
}
//Le reste des valeurs des mois
else {
    valide = 0;
}
if(valide == 0) printf ("Date invalide");
else printf ("La nouvelle date est : %d/%d/%d",j,m,a);
getch();
}

```

### 3-21) Exercice 21-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int valide = 1;
    int j,m,a;
    //Saisir une date
    printf ("Saisir une date (JJ/MM/AA) :");
    scanf ("%d/%d/%d",&j,&m,&a);
    // Calcul de la nouvelle date
    //Les mois de 30 jours et les mois avant de 31 jours
    if(m==4 || m==6 || m==9 || m==11){
        if(j>1 && j<=30) j=j-1;
        else if (j==1) {
            j=31;
            m=m-1; }
        else valide = 0; }
    }

```

```
//Les mois de 31 jours et les mois avant de 30 jours
else if(m==5 || m==7 || m==10 || m==12){
    if(j>1 && j<=31) j=j-1;
    else if (j==1) {
        j=30;
        m=m-1; }
    else valide = 0; }

//Les mois de 31 jours et les mois avant de 31 jours
else if(m==8){
    if(j>1 && j<=31) j=j-1;
    else if (j==1) {
        j=31;
        m=m-1; }
    else valide = 0; }

//Le mois 1
else if(m==1){
    if(j>1 && j<=31) j=j-1;
    else if (j==1) {
        j=31;
        m=12;
        a=a-1; }
    else valide = 0; }

//Le mois 2
else if(m==2){
    if(a%400==0 || a%4==0 && a%100!=0){
        if(j>1 && j<=29) j=j-1;
        else if (j==1) {
            j=31;
            m=m-1; }
        else valide = 0; }
    else {
        if(j>1 && j<=28) j=j-1;
        else if (j==1) {
            j=31;
            m=m-1; }
        else valide = 0; }}

//Le mois 3
else if(m==3){
    if(j>1 && j<=31) j=j-1;
    else if (j==1) {
        if(a%400==0 || a%4==0 && a%100!=0) j=29;
        else j=28;
        m=m-1; }
    } else valide = 0; }

if(valide == 0) printf ("Date invalide");
else printf ("La nouvelle date est : %d/%d/%d",j,m,a);
getch();}
```

**3-22) Exercice 22-2 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    float a,b;
    printf ("Ce programme resout l'équation ax+b=0\n");
    //Saisie de a et b
    printf ("Saisir la valeur de a : ");
    scanf ("%f",&a);
    printf ("Saisir la valeur de b : ");
    scanf ("%f",&b);
    // Calculer la solution si elle existe
    if(a!=0) printf ("x=%f",-b/a);
    else if(a==0 && b!=0)printf("Pas de solution");
    else printf ("Toute valeur est une solution");
    getch();
}
```

**3-23) Exercice 23-3 :**

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main ()
{
    float a,b,c,delta;
    printf ("Ce programme resout l'équation ax^2+bx+c=0\n");
    //Saisie de a et b
    printf ("Saisir la valeur de a : ");
    scanf ("%f",&a);
    printf ("Saisir la valeur de b : ");
    scanf ("%f",&b);
    printf ("Saisir la valeur de c : ");
    scanf ("%f",&c);
    // Calcul de Delta
    delta = b*b-4*a*c;
    // Calcul des solutions si elles existent
    if(a!=0)
    {
        // Cas 1 : delta>0
        if(delta>0){
            printf ("x1=%f\n",(-b-sqrt(delta))/(2*a));
            printf ("x2=%f",(-b+sqrt(delta))/(2*a)); }
```

```

// Cas 2 : delta = 0
else if (delta==0) printf ("x=%.2f",-b/2*a);
// Cas 2 : delta < 0
else if(delta<0) printf ("Pas de solution");
}
else if (b!=0) printf ("x=%.2f",-c/b);
else if (b==0 && c==0) printf ("Toute valeur est une solution");
else if (b==0 && c!=0) printf ("Pas de solution");
getch();
}

```

### 3-24) Exercice 24-3 :

```

#include <stdio.h>
#include <conio.h>
main (){
    //Les opérandes
    float a,b;
    //Variable qui contient l'opération choisie par l'utilisateur
    int op;

    //Affichage du menu
    printf ("1 - Addition\n2 - Soustraction\n3 - Division\n4 -
Multiplication\n");
    // Saisie des nombres
    printf ("Saisir la valeur de a :");
    scanf ("%f",&a);
    printf ("Saisir la valeur de b :");
    scanf ("%f",&b);
    // Choix de l'opération
    printf ("Choisir l'\operation :");
    scanf ("%d",&op);
    //Selon le choix, le programme exécute des instructions bien déterminées
    switch(op){
        case 1 :printf ("La somme est : %.2f",a+b);break;
        case 2 :printf ("La difference est : %.2f",a-b);break;
        case 3 :if (b!=0) printf ("Le résultat de la division est : %.2f",a/b);
                  else printf ("Erreur de division par 0");break;
        case 4 : printf ("Le produit est : %.2f",a*b); break;
        default : printf ("operation invalide"); break;
    }
    getch();
}

```

#### 4. Les instructions de contrôle - Les boucles (for)

##### 4-1) Exercice 1-4 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int i;
    for (i=1;i<=10;i++){
        printf ("Bonjour Mr QARA\n");
    }
    getch();
}
```

##### 4-2) Exercice 2-4 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int i;
    for (i=1;i<=10;i++)
    {
        printf ("%d\n",i);
    }
    getch();
}
```

##### 4-3) Exercice 3-4 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int i;
    for (i=2;i<=20;i=i+2){
        printf ("%d\n",i);
    }
    getch();
}
```

```
#include <stdio.h>
#include <conio.h>
main (){
    int i;
    for (i=1;i<=20;i++) {
        if(i%2==0)
            printf ("%d\n",i);
    }
    getch();
}
```

**4-4) Exercice 4-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,a;
    for (i=1;i<=10;i++){
        printf ("Saisir un entier :");
        scanf ("%d",&a);
    }
    getch();}
```

**4-5) Exercice 5-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,S1=0;
    for (i=1;i<=100;i++){
        S1=S1+i;
    }
    printf("la somme S1 = %d",S1);
    getch();}
```

**4-6) Exercice 6-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,S2=0;
    for (i=1;i<=100;i++){
        if(i%2!=0)
            S2=S2+i;}
    printf("la somme S2 = %d",S2);
    getch();
}
```

**4-7) Exercice 7-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,S3=0;
    for (i=1;i<=100;i++){
        S3=S3+i*i;}
    printf("la somme S3 = %d",S3);
    getch();
}
```

**4-8) Exercice 8-4 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int i;
    float S4=0;
    for (i=1;i<=99;i++)
    {
        S4=S4+(float)i/(i+1);
    }
    printf("la somme S4 = %.2f",S4);
    getch();
}
```

**4-9) Exercice 9-4 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int i;
    float S5=0;
    for (i=1;i<=99;i++){
        if(i%2!=0)
            S5=S5+(float)i/(i+1);
        else
            S5=S5-(float)i/(i+1);
    }
    printf("la somme S5 = %.2f",S5);
    getch();
}
```

**4-10) Exercice 10-4 :**

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int i, P1=1;
    for (i=1;i<=10;i++){
        P1 = P1 * i;
    }
    printf("le produit = %d",P1);
    getch();
}
```

**4-11) Exercice 11-4 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    int i;
    float P1=1;
    for (i=1;i<=19;i=i+2){
        P1 = P1 *(float)i/(i+1);
    }
    printf("le produit = %f",P1);
    getch();
}
```

**4-12) Exercice 12-4 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    int i, p=1;
    for (i=1;i<=11;i++){
        printf ("2^%d = %d\n",i-1,p);
        p=p*2;
    }
    getch();
}
```

**4-13) Exercice 13-4 :**

```
#include <stdio.h>
#include <conio.h>

main ()
{
    int i, n;
    printf ("Saisir un entier positif :");
    scanf ("%d",&n);
    for (i=1;i<=n/2;i++){
        if(n%i==0) printf ("%d\t",i);
    }
    printf ("%d",n);
    getch();
}
```

**4-14) Exercice 14-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i, n;
    int isprime = 1;
    printf ("Saisir un entier positif :");
    scanf ("%d",&n);
    for (i=2;i<=n/2;i++){
        if(n%i==0) isprime = 0;
    }
    if(!isprime) printf ("%d est non premier",n);
    else printf ("%d est premier",n);
    getch();
}
```

**4-15) Exercice 15-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i, n,a,p=1;
    printf ("Saisir un entier a : ");
    scanf ("%d",&a);
    printf ("Saisir un entier positif n : ");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        p = p * a;
    }
    printf ("%d^%d = %d",a,n,p);
    getch();
}
```

**4-16) Exercice 16-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,max,a;
    printf ("Saisir dix entiers :\n");
    for (i=1;i<=10;i++){
        printf ("%d ",i);
        scanf ("%d",&a);
        if(i==1 || max<a) max=a;
    }
    printf ("Le maximum est : %d",max);
    getch();
}
```

**4-17) Exercice 17-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,n,u=1;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        u=2*u+5;
    }
    printf ("U(%d)=%d",n,u);
    getch();
}
```

**4-18) Exercice 18-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,n,u=1,v=1,t;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=2;i<=n;i++){
        t=u;
        u=u+2*v+3;
        v=t;
    }
    printf ("U(%d)=%d",n,u);
    getch();
}
```

**4-19) Exercice 19-4 :**

```
#include <stdio.h>
#include <conio.h>
main (){
    int i,j,n;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        for (j=1;j<=i;j++)
            printf ("+");
        printf ("\n");
    }
    getch();
}
```

```
main (){
    int i,j,n;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        for (j=1;j<=n-i+1;j++)
            printf ("+");
        printf ("\n");
    }
    getch();
}
```

#### 4-20) Exercice 20-4 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int i,j,n;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        for (j=1;j<=i-1;j++)
            printf (" ");
        for (j=1;j<=n-i+1;j++)
            printf ("+");
        printf ("\n");
    }
    getch();
}

main (){
    int i,j,n;
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    for (i=1;i<=n;i++){
        for (j=1;j<=n-i;j++)
            printf (" ");
        for (j=1;j<=i;j++)
            printf ("+");
        printf ("\n");
    }
    getch();
}
```

## 5. Les instructions de contrôle - Les boucles (while et do while)

### 5-1) Exercice 1-5 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int i=1;
    while (i<=10){
        printf ("Bonjour Mr QARA\n");
        i++;
    }
    getch();}
```

### 5-2) Exercice 2-5 :

```
#include <stdio.h>
#include <conio.h>
main (){
    float a;
    do {
        printf ("saisir un nombre positif :");
        scanf ("%f",&a);
    } while (a<=0);
    getch();}
```

```
#include <stdio.h>
#include <conio.h>
main (){
    float a;
    printf ("saisir un nombre positif :");
    scanf ("%f",&a);
    while (a<=0){
        printf ("saisir un nombre positif :");
        scanf ("%f",&a);
    }
    getch();
}
```

### 5-3) Exercice 3-5 :

```
#include <stdio.h>
#include <conio.h>
main (){
    float a;
    do {
        printf ("saisir un nombre positif :");
        scanf ("%f",&a);
        if(a<0) printf ("Plus grand\n");
    }
```

```

else if (a>20) printf (" plus petit"),
} while (a<0 || a>20);
getch();}

```

#### 5-4) Exercice 4-5 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int i,a,p=1;
    int j=0;
    printf ("saisir entier positif :");
    scanf ("%d",&a);
    do {
        p=p*2;
        j++;
    } while (p<a);
    printf ("2^%d < %d <= 2^%d",j-1,a,j);
    getch();}

```

#### 5-5) Exercice 5-5 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int i,a,m=0;
    printf ("saisir entier positif :");
    scanf ("%d",&a);
    do {
        m=m+5;
    } while (m<a);
    printf ("%d < %d <= %d",m-5,a,m);
    getch();}

```

#### 5-6) Exercice 6-5 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int i=2,a;
    printf ("saisir entier positif :");
    scanf ("%d",&a);
    while (a%i!=0 && i<=a/2) i++;
    if(i<=a/2) printf ("%d n'est pas premier",a);
    else printf ("%d est premier",a);
    getch();
}

```

## 5-7) Exercice 7-2 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float a,s=0,i=0;
    do {
        printf ("saisir un nombre : ");
        scanf ("%f",&a);
        if(a>0) {
            s=s+a;
            i=i+1;}
        } while (a!=0);
    printf ("La somme des nombres positifs : %.2f\n",s);
    printf ("La moyenne des nombres positifs : %.2f",s/i);
    getch();
}
```

## 5-8) Exercice 8-5 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float a,s=0;
    int i=0;
    do {
        printf ("saisir un nombre : ");
        scanf ("%f",&a);
        if(a>0) {
            i=i+1;
            }
        } while (a!=0);
    printf ("%d nombres positifs\n",i);
    getch();
}
```

## 5-9) Exercice 9-5 :

```
#include <stdio.h>
#include <conio.h>
main () {
    int a,b,t;
    printf ("saisir un entier positif a : ");
    scanf ("%d",&a);
    printf ("saisir un entier positif b : ");
    scanf ("%d",&b);
```

```
//Sauvegarder les valeurs de a et b pour l'affichage du résultat
int a1 = a;
int b1 = b;
while (b!=0) {
    t=a;
    a=b;
    b=t%b;}
printf ("PGCD (%d , %d ) = %d",a1,b1,a);
getch();
}
```

### 5-10) Exercice 10-5 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int a,b,q=0;
    printf ("saisir un entier positif a : ");
    scanf ("%d",&a);
    printf ("saisir un entier positif b : ");
    scanf ("%d",&b);
    //Sauvegarder les valeurs de a et b pour l'affichage du résultat
    int a1 = a;
    int b1 = b;
    while (a>0) {
        q=q+1;
        a=a-b;
    }
    printf ("%d / %d = %d\n",a1,b1,q-1);
    printf ("%d mod %d = %d\n",a1,b1,a+b);
    getch();
}
```

### 5-11) Exercice 11-5 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int a,b,t;
    printf ("saisir un entier positif a : ");
    scanf ("%d",&a);
    printf ("saisir un entier positif b : ");
    scanf ("%d",&b);
    //Sauvegarder les valeurs de a et b pour l'affichage du résultat
    int a1 = a;
    int b1 = b;
```

```

//Calcul du PGCD
while (b!=0) {
    t=a;
    a=b;
    b=t%b;
}
//Si Pgcd (a,b)=1 ils sont premiers entre eux
if(a==1) printf ("%d et %d sont premiers entre eux",a1,b1);
else printf("%d et %d ne sont pas premiers entre eux",a1,b1);
getch();
}

```

### 5-12) Exercice 12-5 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int a,r;
    printf ("saisir un nombre octal : ");
    scanf ("%d",&a);
    do{
        r=a%10;
        a=a/10;
    }
    while (r<=7 && a!=0);
    if(r>7) printf ("Ce n'est pas un nombre octal");
    else printf ("C'est un nombre octal");
    getch();
}

```

### 5-13) Exercice 13-5 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int a,r;
    printf ("saisir un nombre binaire : ");
    scanf ("%d",&a);
    do{
        r=a%10;
        a=a/10;
    }
    while (r<=1 && a!=0);
    if(r>1) printf ("Ce n'est pas un nombre binaire");
    else printf ("C'est un nombre binaire");
    getch();
}

```

## 5-14) Exercice 14-5 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int a,r,o=0,p=1;
    printf ("saisir un nombre decimal : ");
    scanf ("%d",&a);
    do{
        r=a%8;
        o = o + r * p;
        a=a/8;
        p = p * 10;
    }
    while (a!=0);
    printf ("Le resultat en octal : %d",o);
    getch();
}
```

## 5-15) Exercice 15-5 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int a,r,b=0,p=1;
    printf ("saisir un nombre decimal : ");
    scanf ("%d",&a);
    do{
        r=a%2;
        b = b + r * p;
        a=a/2;
        p = p * 10;
    }
    while (a!=0);
    printf ("Le resultat en binaire : %d",b);
    getch();
}
```

## 5-16) Exercice 16-5 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int a,r,d=0,p=1;
    printf ("saisir un nombre binaire : ");
    scanf ("%d",&a);
```

```
do{
    r=a%10;
    d = d + r * p;
    a=a/10;
    p = p * 2;
}
while (a!=0);
printf ("Le resultat en decimal : %d",d);
getch();
```

### 5-17) Exercice 17-5 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int a,r,d=0,p=1;
    printf ("saisir un nombre octal : ");
    scanf ("%d",&a);
    do{
        r=a%10;
        d = d + r * p;
        a=a/10;
        p = p * 8;
    }
    while (a!=0);
    printf ("Le resultat en decimal : %d",d);
    getch();
}
```

## 6. La programmation modulaire : les fonctions

### 6-1) Exercice 1-6 :

```
#include <stdio.h>
#include <conio.h>

// Fonction Produit
float Produit (float a, float b)
{
    float p;
    p=a*b;
    return p;
}
main ()
{
    float x,y;
    printf ("Saisir un entier :");
    scanf ("%f",&x);
    printf ("Saisir un entier :");
    scanf ("%f",&y);
    printf ("Le produit est : %.2f",Produit(x,y));
    getch();
}
```

### 6-2) Exercice 2-6 :

```
#include <stdio.h>
#include <conio.h>

// Fonction SurfaceCercle
float SurfaceCercle ( float r)
{
    float s;
    s=3.14*r*r;
    return s;
}

main ()
{
    float r;
    printf ("Saisir le rayon du cercle :");
    scanf ("%f",&r);
    printf ("La surface du cercle est : %.2f",SurfaceCercle(r));
    getch();
}
```

**6-3) Exercice 5-0 :**

```
#include <stdio.h>
#include <conio.h>
// Fonction Echange
void Echange (float *a, float *b)
{
    float t;
    t=*a;
    *a=*b;
    *b=t;
}
main ()
{
    float x,y;
    printf ("Saisir un nombre :");
    scanf ("%f",&x);
    printf ("Saisir un nombre :");
    scanf ("%f",&y);
    Echange (&x,&y);
    printf ("Apres l'echange : \nx=%f\ty=%f",x,y);
    getch();
}
```

**6-4) Exercice 4-6 :**

```
#include <stdio.h>
#include <conio.h>
// Fonction SiPair
int SiPair (int a) {
    if(a%2==0) return 1;
    else return -1;
}

main ()
{
    //Déclaration d'un nombre entier
    int a;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    // Tester si le nombre est pair ou impair
    if(SiPair(a)==1) printf ("%d est pair",a);
    else printf ("%d est impair",a);
    getch();
}
```

**6-5) Exercice 5-0 :**

```
#include <stdio.h>
#include <conio.h>
// Fonction Signe
int Signe (float x)
{
    if(x>0) return 1;
    else if(x<0) return -1;
    else return 0;
}
main (){
    //Déclaration d'un nombre entier
    float a;
    printf ("Saisir un entier :");
    scanf ("%f",&a);
    // Tester si le nombre est positif, négatif ou nul
    if(Signe(a)>0) printf ("% .2f est positif",a);
    else if (Signe(a)<0)printf ("% .2f est negatif",a);
    else printf ("% .2f est nul",a);
    getch();
}
```

**6-6) Exercice 6-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction Multiple vérifie si x est un multiple de y
int Multiple (int x, int y)
{
    if(x%y==0) return 1;
    else return 0;
}
main ()
{
    //Déclaration d'un nombre entier
    int a,n;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    // Tester si a est un multiple de n ou non
    if(Multiple(a,n)==1) printf ("%d est un multiple de %d",a,n);
    else printf ("%d n'est pas un multiple de %d",a,n);
    getch();
}
```

**6-7) Exercice 7-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction Diviseur vérifie si x est un diviseur de y
int Diviseur (int x, int y) {
    if(y%x==0) return 1;
    else return 0;
}
main ()
{
    //Déclaration d'un nombre entier
    int a,n;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    printf ("Saisir un entier :");
    scanf ("%d",&n);
    // Tester si a est un diviseur de n ou non
    if(Diviseur(a,n)==1) printf ("%d est un diviseur de %d",a,n);
    else printf ("%d n'est pas un diviseur de %d",a,n);
    getch();
}
```

**6-8) Exercice 8-6 :**

```
#include <stdio.h>
#include <conio.h>

// La fonction Max
float Max (float x, float y)
{
    if(x>y) return x;
    else return y;
}

main ()
{
    float a,b;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    printf ("Saisir un nombre :");
    scanf ("%f",&b);
    // Comparer les deux nombres pour afficher le maximum
    printf ("Le maximum est : %.2f", Max(a,b));
    getch();
}
```

```
#include <stdio.h>
#include <conio.h>
// La fonction Max
float Max (float x, float y){
    if(x>y) return x;
    else return y;
}
main (){
    float a,b,c;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    printf ("Saisir un nombre :");
    scanf ("%f",&b);
    printf ("Saisir un nombre :");
    scanf ("%f",&c);
    // Comparer les trois nombres pour afficher le maximum
    printf ("Le maximum est : %.2f", Max(Max(a,b),c));
    getch();
}
```

### 6-9) Exercice 9-6 :

```
#include <stdio.h>
#include <conio.h>
// La fonction inverse
float Inverse (float x){
    if(x!=0) return 1/x;
    else return 0;
}
main (){
    float a;
    printf ("Saisir un nombre :");
    scanf ("%f",&a);
    if(Inverse(a)!=0) printf ("L'inverse est : %f",Inverse(a));
    else printf ("Erreur de division par 0");
    getch();
}
```

### 6-10) Exercice 10-6 :

```
#include <stdio.h>
#include <conio.h>
// La fonction Message
void Message (int n)
{
    int i;
    for (i=1;i<=n;i++)
        printf ("Bonjour Monsieur QARA\n");
}
```

```
main (){
    // Appel de la fonction
    Message (10);
    getch();
}
```

### 6-11) Exercice 11-6 :

```
#include <stdio.h>
#include <conio.h>
// La fonction somme
int somme (int n)
{
    int i,S=0;
    for (i=1;i<=n;i++) S=S+i;
    return S;
}
main ()
{
    int n;
    printf ("Saisir un entier positif :");
    scanf("%d",&n);
    printf("S(%d) = %d",n,somme(n));
    getch();
}
```

### 6-12) Exercice 12-6 :

```
#include <stdio.h>
#include <conio.h>
int Fact (int n)
{
    int i, P=1;
    for (i=1;i<=n;i++) P=P*i;
    return P;
}

main ()
{
    int n;
    printf ("Saisir un entier positif :");
    scanf("%d",&n);
    printf("%d! = %d",n,Fact(n));
    getch();
}
```

**6-13) Exercice 13-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction SiPremier
int SiPremier (int n)
{
    int i;
    int isprime = 1;
    for (i=2;i<=n/2;i++){
        if(n%i==0) isprime = 0;
    }
    return isprime;
}
main (){
    int n;
    printf ("Saisir un entier positif :");
    scanf ("%d",&n);
    if(!SiPremier(n)) printf ("%d est non premier",n);
    else printf ("%d est premier",n);
    getch();
}
```

**6-14) Exercice 14-6 :**

```
#include <stdio.h>
#include <conio.h>

// La fonction Puissance
int Puissance (int x, int y)
{
    int i,p=1;
    for (i=1;i<=y;i++) p=p*x;
    return p;
}
main ()
{
    int a,n;
    printf ("Saisir un entier a : ");
    scanf ("%d",&a);
    printf ("Saisir un entier positif n : ");
    scanf ("%d",&n);
    printf ("%d^%d = %d",a,n,Puissance(a,n));
    getch();
}
```

**6-15) Exercice 15-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction Etoile
void Etoile (int n){
    int i,j;
    for (i=1;i<=n;i++){
        for (j=1;j<=i;j++)
            printf ("+");
        printf ("\n");}
main (){
    int i,j,n;
    printf ("Saisir un nombre :");
    scanf ("%d",&n);
    Etoile (n);
    getch();}
```

**6-16) Exercice 16-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction LirePositif
void LirePositif (float *x){
    do {
        printf ("saisir un nombre positif :");
        scanf ("%f",x);
    } while (*x<=0);}
main (){
    float a;
    LirePositif(&a);
    printf ("%.2f",a);
    getch();
}
```

**6-17) Exercice 17-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction PGCD
int PGCD (int a, int b){
    int t;
    while (b!=0) {
        t=a;
        a=b;
        b=t%b;
    }
    return a;}
```

```

main ()
{
    int a,b,t;
    printf ("saisir un entier positif a : ");
    scanf ("%d",&a);
    printf ("saisir un entier positif b : ");
    scanf ("%d",&b);
    printf ("PGCD (%d , %d ) = %d",a,b,PGCD(a,b));
    getch();
}

```

### 6-18) Exercice 18-6 :

```

#include <stdio.h>
#include <conio.h>
// La fonction DecBin
int DecBin (int a){
    int r,o=0,p=1;
    do{
        r=a%2;
        o = o + r * p;
        a=a/2;
        p = p * 10;
    }
    while (a!=0);
    return o;
}
main (){
    int x;
    printf ("saisir un nombre decimal : ");
    scanf ("%d",&x);
    printf ("Le resultat en binaire : %d",DecBin(x));
    getch();
}

```

### 6-19) Exercice 19-6 :

```

#include <stdio.h>
#include <conio.h>
// La fonction SiBinaire
int SiBinaire (int x) {
    int r;
    do {
        r=x%10;
        x=x/10;
    }
    while (r<=1 && x!=0);
    if(r>1) return 0;
    else return 1;
}

```

```

main (){
    int a;
    printf ("saisir un nombre binaire : ");
    scanf ("%d",&a);
    if(!SiBinaire(a)) printf ("Ce n'est pas un nombre binaire");
    else printf ("C'est un nombre binaire");
    getch();
}

```

### 6-20) Exercice 20-6 :

```

#include <stdio.h>
#include <conio.h>
// La fonction DecOct
int DecOct (int a)
{
    int r,o=0,p=1;
    do{
        r=a%8;
        o = o + r * p;
        a=a/8;
        p = p * 10;
    }
    while (a!=0);
    return o;
}
main (){
    int x;
    printf ("saisir un nombre decimal : ");
    scanf ("%d",&x);
    printf ("Le resultat en octal : %d",DecOct(x));
    getch();
}

```

### 6-21) Exercice 21-6 :

```

#include <stdio.h>
#include <conio.h>
// La fonction DecBin
int DecBin (int a){
    int r,o=0,p=1;
    do{
        r=a%2;
        o = o + r * p;
        a=a/2;
        p = p * 10;
    } while (a!=0);
    return o;
}

```

```
main (){
    int x;
    printf ("saisir un nombre decimal : ");
    scanf ("%d",&x);
    printf ("Le resultat en binaire : %d",DecBin(x));
    getch();
}
```

**6-22) Exercice 22-6 :**

```
#include <stdio.h>
#include <conio.h>
// La fonction DecBin
int PGCD (int a,int b)
{
    if(b!=0) return PGCD (b,a%b);
    else return a;
}
main (){
    int x,y,p;
    printf ("saisir un nombre : ");
    scanf ("%d",&x);
    printf ("saisir un nombre : ");
    scanf ("%d",&y);
    printf ("pgcd = %d",PGCD(x,y));
    getch();
}
```

## 7. Les tableaux

### 7-1) Exercice 1-7 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float T[10];
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%f",&T[i]);
    }
    //retour à la ligne
    printf ("\nAffichage du tableau:\n");
    //L'affichage
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = %.2f\n",i,T[i]);
    }
    getch();
}
```

### 7-2) Exercice 2-7 :

```
#include <stdio.h>
#include <conio.h>
main (){
    float T[10],max;
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%f",&T[i]);
    }
    //Calcul de la note maximale
    max=T[0];
    for (i=0;i<10;i++)
    if(max<T[i]) max=T[i];
    //L'affichage de la valeur maximale
    printf ("La valeur maximale est : %.2f",max);
    getch();
}
```

## 7-3) Exercice 5-7 :

```
#include <stdio.h>
#include <conio.h>
main (){
    float T[10],n;
    int i,existe=0;
    printf ("Saisie du tableau :\n");
    // La saisie du tableau
    for (i=0;i<10;i++){
        printf ("T[%d] = ",i);
        scanf("%f",&T[i]);
    }
    // La saisie du nombre à chercher
    printf ("Saisir un nombre : ",i);
    scanf("%f",&n);
    //Chercher dans le tableau
    for (i=0;i<10;i++)
        if(n==T[i]) existe = 1;
    if(existe)
        printf ("La valeur %.2f existe dans le tableau");
    else
        printf ("La valeur %.2f n'existe pas dans le tableau");
    getch();
}
```

## 7-4) Exercice 4-7 :

```
#include <stdio.h>
#include <conio.h>
main (){
    float T[10];
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie du tableau
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%f",&T[i]);
    }
    //retour à la ligne
    printf ("\nAffichage des nombres positifs:\n");
    //L'affichage
    for (i=0;i<10;i++)
        if (T[i]>=0)printf ("T[%d] = %.2f\n",i,T[i]);
    getch();
}
```

**7-5) Exercice 5- :** 

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10];
    int i,nbr=0;
    printf ("Saisie du tableau :\n");
    // La saisie du tableau
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
    }
    //retour à la ligne
    printf ("\nAffichage des nombres pairs:\n");
    //L'affichage des nombres pairs
    for (i=0;i<10;i++)
    {
        if (T[i]%2==0)
        {
            printf ("T[%d] = %d\n",i,T[i]);
            nbr++;
        }
    }
    printf ("%d nombres pairs",nbr);
    getch();
}
```

**7-6) Exercice 6-7 :** 

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int a,n, i, r, nbr=0;
    printf ("Saisir un entier :");
    scanf ("%d",&a);
    n=a;
    // Calcul de la taille du tableau
    do{
        a=a/2;
        nbr++;
    }
    while (a!=0);
    // Déclaration du tableau
```

```

int bin[nbr];
/* Calcul des restes et remplissage
du tableau de la fin au début */
i=nbr-1;
do {
    r=n%2;
    n=n/2;
    bin[i]=r;
    i--;
}
while (n!=0);
// Affichage du résultat
for (i=0;i<nbr;i++) printf ("%d",bin[i]);
getch();
}

```

### 7-7) Exercice 7-7 :

```

#include <stdio.h>
#include <conio.h>
main ()
{
    int T1[10], T2[10];
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie du tableau
    for (i=0;i<10;i++)
    {
        printf ("T1[%d] = ",i);
        scanf("%d",&T1[i]);
    }
    // Inverser le tableau
    for (i=0;i<10;i++)
        T2[9-i]=T1[i];
    // Affichage du nouveau tableau
    printf ("\nAffichage du nouveau tableau: \n");
    for (i=0;i<10;i++)
        printf ("T2[%d] = %d\n",i,T2[i]);
    getch();
}

```

```

#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10],t;
    int i;
    printf ("Saisie du tableau :\n");

```

```

// La saisie du tableau
for (i=0;i<10;i++)
{
    printf ("T[%d] = ",i);
    scanf("%d",&T[i]);
}
// Inverser le tableau
for (i=0;i<10/2;i++)
{
    t=T[9-i];
    T[9-i]=T[i];
    T[i]=t;
}
// Affichage du nouveau tableau
printf ("\nAffichage du nouveau tableau: \n");
for (i=0;i<10;i++)
    printf ("T[%d] = %d\n",i,T[i]);
getch();
}

```

### 7-8) Exercice 8-7 :

```

#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10];
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
    }

    // Remplacer les nombres négatifs par des 0
    for (i=0;i<10;i++)
        if(T[i]<0) T[i]=0;
    //retour à la ligne
    printf ("\nAffichage du tableau:\n");
    //L'affichage
    for (i=0;i<10;i++)
        printf ("T[%d] = %d\n",i,T[i]);
    getch();
}

```

## 7-9) Exercice 9- : :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10], s=0;
    float moy;
    int i;
    printf ("Saisie du tableau :\n");
    // La saisie et le calcul
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
        s=s+T[i];
    }
    moy =(float) s / 10;
    printf ("\nLa somme est : %d\nLa moyenne est : %.2f\n",s,moy);
    getch();
}
```

## 7-10) Exercice 10-7 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10],t;
    int i,k,dim=10;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
    }
    // Supprimer les nombres négatifs
    i=0;
    while (i<dim) {
        if (T[i]<0) {
            dim--;
            int j;
            for(j=i;j<dim;j++) T[j]=T[j+1];
        }
        else i++;
    }
    //retour à la ligne
```

```

printf ("\nAffichage du tableau.\n"),
//L'affichage
for (i=0;i<dim;i++)
{
    printf ("T[%d] = %d\n",i,T[i]);
}
getch();
}

```

### 7-11) Exercice 11-7 :

```

#include <stdio.h>
#include <conio.h>
main ()
{
    int T[10];
    int i,x=0,temp;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
    }
    // Tri du tableau
    while(x==0)
    {
        int temp;
        x = 1;
        for(i=0 ; i<9 ; i++)
        {
            if(T[i] > T[i+1])
            {
                temp = T[i];
                T[i]= T[i+1];
                T[i+1]=temp;
                x=0;
            }
        }
    }
    printf ("\nAffichage du tableau:\n");
    //L'affichage
    for (i=0;i<10;i++)
        printf ("T[%d] = %d\n",i,T[i]);
    getch();
}

```

### 7-12) Exercice 12-7 :

```
#include <stdio.h>
#include <conio.h>
main () {
    int T[10];
    int i,x=0,temp;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++) {
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]); }
    // Tri du tableau
    while(x==0)
    {
        int temp;
        x = 1;
        for(i=0 ; i<9 ; i++)
        {
            if(T[i] < T[i+1])
            {
                temp = T[i];
                T[i]= T[i+1];
                T[i+1]=temp;
                x=0;
            }
        }
    }
    printf ("\nAffichage du tableau:\n");

    //L'affichage
    for (i=0;i<10;i++)
    {
        printf ("T[%d] = %d\n",i,T[i]);
    }
    getch();
}
```

### 7-13) Exercice 13-7 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int T[10];
    int i,j,temp,min;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++){
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]); }
```

```

// Tri du tableau
for(i=0;i<9;i++){
    min=i;
    for(j=i+1;j<=9;j++){
        if(T[j]<T[min]) min=j; }
    if(min!=i) {
        temp = T[i];
        T[i]=T[min];
        T[min]=temp;
    }
}
printf ("\nAffichage du tableau:\n");
//L'affichage
for (i=0;i<10;i++)
{
    printf ("T[%d] = %d\n",i,T[i]);
}
getch();}
```

### 7-14) Exercice 14-7 :

```

#include <stdio.h>
#include <conio.h>
main (){
    int T[10];
    int i,j,temp,min;
    printf ("Saisie du tableau :\n");
    // La saisie
    for (i=0;i<10;i++){
        printf ("T[%d] = ",i);
        scanf("%d",&T[i]);
    }
    // Tri du tableau
    for(i=0;i<9;i++){
        min=i;
        for(j=i+1;j<=9;j++){
            if(T[j]>T[min]) min=j;
        }
        if(min!=i) {
            temp = T[i];
            T[i]=T[min];
            T[min]=temp;}}
    //L'affichage
    printf ("\nAffichage du tableau:\n");
    for (i=0;i<10;i++) {
        printf ("T[%d] = %d\n",i,T[i]);
    }
    getch();}
```

## 7-15) Exercice 15-1 :

```
#include <stdio.h>
#include <conio.h>
main ()
{
    float T[10], s=0, moy;
    int i, j, x, temp;
    printf ("Saisie du tableau :\n");
    // La saisie et le calcul
    for (i=0; i<10; i++)
    {
        printf ("T[%d] = ", i);
        scanf ("%f", &T[i]);
        s=s+T[i];
    }

    // Tri du tableau par ordre de mérite
    while(x==0)
    {
        int temp;
        x = 1;
        for(i=0 ; i<9 ; i++)
        {
            if(T[i] < T[i+1])
            {
                temp = T[i];
                T[i]= T[i+1];
                T[i+1]=temp;
                x=0;
            }
        }
    }

    //Calcul et affichage de la moyenne
    moy = s / 10;
    printf ("La moyenne de la classe est : %.2f\n", moy);
    printf ("La note du majorant est : %.2f\n", T[0]);
    printf ("\nAffichage des notes >= %.2f:\n", moy);
    //L'affichage
    for (i=0; i<10; i++)
    {
        if (T[i]>=moy)printf ("T[%d] = %.2f\n", i, T[i]);
    }
    getch();
}
```

## 7-16) Exercice 16-1 :

```
#include <stdio.h>
#include <conio.h>
main () {
    int T[5][3];
    int i,j,max, nbr;
    printf ("Saisie du tableau :\n");
    // La saisie et les calculs
    for (i=0;i<5;i++) {
        for (j=0;j<3;j++){
            printf ("T[%d][%d] = ",i,j);
            scanf("%d",&T[i][j]);
            if(T[i][j]>=0) nbr++;
            if(i==0 && j==0 || max<T[i][j]) max=T[i][j]; }
    }
    printf ("\nLe maximum est : %d\n",max);
    printf ("\nLe nombre des entiers positifs est : %d\n",nbr);
    printf ("\n\nAffichage du tableau:\n");
    //L'affichage
    for (i=0;i<5;i++) {
        for (j=0;j<3;j++)
            printf ("%4d\t",T[i][j]);
        printf("\n");
    }
    getch();
}
```

## 7-17) Exercice 17-7 :

```
#include <stdio.h>
#include <conio.h>
main (){
    int T[4][4];
    int i,j,tr=0;

    printf ("Saisie du tableau :\n");
    // La saisie et les calculs
    for (i=0;i<3;i++) {
        for (j=0;j<3;j++){
            printf ("T[%d][%d] = ",i,j);
            scanf("%d",&T[i][j]);
            if(i==j) tr=tr+T[i][j];
        } }
    printf ("\nLa trace est : %d\n",tr);
    getch();
}
```

## 8. Les chaines de caractères

### 8-1) Exercice 1-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char nom[10];
    printf ("Saisir votre nom: ");
    gets(nom);
    printf ("Votre nom est : ");
    puts(nom);
    printf ("La longeur de votre nom est : %d",strlen(nom));
    getch();
}
```

### 8-2) Exercice 2-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char nom[10];
    char ch[10];
    printf ("Saisir votre nom: ");
    gets(nom);
    strcpy(ch,nom);
    printf ("Votre nom est : ");
    puts(ch);
    getch();
}
```

### 8-3) Exercice 3-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char fullname[10];
    char *ch;
    printf ("Saisir votre nom complet : ");
    gets(fullname);
    ch=strchr (fullname,' ');
    printf ("Votre nom est : ");
    puts(ch+1);
    getch();}
```

**8-4) Exercice 4-8 :**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main ()
{
    char ch1[10];
    char ch2[10];
    printf ("Saisir une chaîne ch1 : ");
    gets(ch1);
    printf ("Saisir une chaîne ch2 : ");
    gets(ch2);
    if(strcmp(ch1,ch2)<0){
        puts(ch1);
        puts(ch2);
    }
    else {
        puts(ch2);
        puts(ch1);
    }
    getch();
}
```

**8-5) Exercice 5-8 :**

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main ()
{
    char ch1[10];
    char ch2[10];
    printf ("Saisir une chaîne ch1 : ");
    gets(ch1);
    printf ("Saisir une chaîne ch2 : ");
    gets(ch2);
    if(stricmp(ch1,ch2)<0)
    {
        puts(ch1);
        puts(ch2);
    }
    else {
        puts(ch2);
        puts(ch1);
    }
    getch();}
```

### 8-6) Exercice 6-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char ch1[10];
    char ch2[10];
    char *ch;
    printf ("Saisir une chaîne ch1 : ");
    gets(ch1);
    printf ("Saisir une chaîne ch2 : ");
    gets(ch2);
    ch=strcat(ch1," ");
    ch=strcat(ch,ch2);
    puts(ch);
    getch();
}
```

### 8-7) Exercice 7-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char ch1[10];
    char ch2[10];
    int i;
    printf ("Saisir une chaîne ch1 : ");
    gets(ch1);
    for (i=0;i<strlen(ch1);i++) ch2[i]=ch1[strlen(ch1)-1-i];
    puts(ch2);
    getch();
}
```

### 8-8) Exercice 8-8 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char ch[10];
    char ch2[10];
    int i;
    printf ("Saisir une chaîne ch : ");
    gets(ch);
    for (i=0;i<strlen(ch);i++) ch2[i]=ch[strlen(ch)-1-i];
    if(strcmp(ch,ch2)==0) printf ("La chaîne est un palindrome");
```

```

else printf ("La chaine n'est pas un mot terminé");
getch();
}

```

### 8-9) Exercice 9-8 :

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
main (){
    char *ch;
    char *esp;
    esp = calloc(100,sizeof(char));
    ch = calloc(100,sizeof(char));
    int nbr=0;
    printf ("Saisir une chaine : ");
    gets(ch);
    strcpy(esp,ch);
    // Calcul du nombre des mots
    do {
        while ((*esp)==' ') esp++;
        if (*esp]!='\0') nbr++;
        esp=strchr(esp,' ');
    }
    while (esp!=NULL);
    printf ("Nombre de mots : %d",nbr);
    getch();
}

```

### 8-10) Exercice 10-8 :

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
main () {
    char *ch;
    char c;
    int i,dim;
    ch = calloc(100,sizeof(char));
    printf ("Saisir une chaine : ");
    gets(ch);
    dim=strlen(ch);
    printf ("Saisir une caractere : ");
    c=getchar ();
    i=0;
    while (i<dim){

```

```

if (ch[i]==c) {
    dim--;
    int j;
    for(j=i;j<dim;j++) ch[j]=ch[j+1];
    ch[dim]='\0';
} else i++;
puts(ch);
getch();
}

```

### 8-11) Exercice 11-8 :

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
main (){
    char ch[20];
    int i;
    printf ("Saisir une chaîne ch : ");
    gets(ch);
    for (i=0;i<strlen(ch);i++) {
        if(ch[i]>=97 && ch[i]<=122) ch[i]= ch[i]-32;
        else if (ch[i]>=65 && ch[i]<=90) ch[i]= ch[i]+32;
    }
    puts(ch);
    getch();
}

```

### 8-12) Exercice 12-8 :

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
main ()
{
    char ch[20], c1, c2;
    int i;
    printf ("Saisir une chaîne ch : ");
    gets(ch);
    printf ("Saisir un caractère c1 : ");
    c1=getchar();
    while(getchar()!='\n');
    printf ("Saisir un caractère c2 : ");
    c2=getchar();
    for (i=0;i<strlen(ch);i++)
        if(ch[i]==c1) ch[i]= c2;
    puts(ch); getch();
}

```

## 8-13) Exercice 15-0 :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
char* encrypt (char e[],int key) {
    int i;
    char *d;
    d=malloc(sizeof(char),20);
    for (i=0;i<strlen(e);i++) d[i] = e[i] + key;
    return d;}
char* decrypt (char d[],int key) {
    int i;
    char *e;
    e=malloc(sizeof(char),20);
    for (i=0;i<strlen(d);i++) e[i] = d[i] - key;
    return e;
}
main (){
    char *e, *d;
    e=malloc(sizeof(char),20);
    d=malloc(sizeof(char),20);
    int k;
    puts ("Saisir un message en clair : ");
    gets(e);
    puts ("Saisir un entier (the key) : ");
    scanf("%d",&k);
    d=encrypt(e,k);
    puts("\nLe message chiffre est : ");
    puts(d);
    puts("\nLe message d'origine est : ");
    puts(decrypt (d,k));
    getch();
}
```

## 9. Les structures

### 9-1) Exercice 1-9 :

```
1) Question 1 :  
// 1) La structure complexe  
struct complexe {  
    float re;  
    float im;  
};  
typedef struct complexe cpx;  
2) Question 2 :  
// 2) Fonction "afficher" pour afficher un nombre complexe  
void afficher (cpx a) {  
    if(a.im>0) printf ("% .2f + % .2f i\n",a.re,a.im); // 5+7 x i  
    else if(a.im<0) printf ("% .2f - % .2f i\n",a.re,-(a.im)); // 5-7 x i  
    else printf ("% .2f",a.re);  
}  
3) Question 3 :  
// 3) Fonction "lire" pour lire un nombre complexe  
void lire (cpx *a) {  
    printf("Saisir un complexe ( x+(y)i ):");  
    scanf ("%f + (%f)i",&a->re,&a->im);  
}  
4) Question 4 :  
// 4) La fonction somme qui permet de calculer la somme de 2 complexes  
cpx somme (cpx a, cpx b){  
    cpx s;  
    s.re = a.re + b.re;  
    s.im = a.im + b.im;  
    return s;  
}  
5) Question 5 :  
// 5) La fonction produit qui permet de calculer le produit de 2 complexes  
cpx produit (cpx a, cpx b){  
    cpx p;  
    p.re = ( a.re * b.re ) - ( a.im * b.im );  
    p.im = ( a.re * b.im ) + ( a.im * b.re );  
    return p;  
}  
6) Question 6 :  
// 6) La fonction conjugué qui permet de calculer le conjugué d'un complexe  
cpx conjugué (cpx a){  
    cpx b;  
    b.re = a.re;  
    b.im = - a.im;  
    return b;  
}
```

7) Question 7 .

```
// 7) La fonction module qui permet de calculer le module d'un complexe
float module (cpx a){
    float b;
    b=sqrt((a.re)*(a.re)+(a.im)*(a.im));
    return b;
}
```

8) Question 8 :

```
// 8) La fonction inverse qui permet de calculer l'inverse d'un complexe
cpx inverse (cpx a){
    cpx b;
    b.re = a.re/(module(a)*module(a));
    b.im = - a.im/(module(a)*module(a));
    return b;}
```

9) Question 9 :

```
// 9) La fonction div qui permet de calculer a/b
cpx div (cpx a, cpx b){
    cpx d;
    d=produit(a,inverse(b));
    return d;}
```

10) Question 10 :

```
main (){
    //Déclaration des variables de type struct complexe (cpx)
    cpx z1,z2,p,s,d;
    //Appel de la fonction lire pour la saisie
    lire(&z1);
    lire(&z2);
    // Calculs
    p=produit(z1,z2);
    s=somme(z1,z2);
    d=div(z1,z2);
    //Affichage des résultats
    printf ("la somme est : ");
    afficher(s);
    printf ("la produit est : ");
    afficher(p);
    printf ("le resultat de la division est : ");
    afficher(d);
    getch();}
```

### 9-2) Exercice 2-9 :

1) Question 1 :

```
// 1) La structure rationnel
typedef struct rationnel {
    int a;
    int b;
} ratio;
```

2) Question 2 .

```
// 2) La fonction somme qui calcule la somme de deux rationnels
ratio somme (ratio x, ratio y) {
    ratio s;
    s.a=(x.a)*(y.b)+(y.a)*(x.b);
    s.b= (x.b)*(y.b);
    return s;}
```

3) Question 3 :

```
// 3) La fonction produit qui calcule le produit de deux rationnels
ratio produit (ratio x, ratio y) {
    ratio p;
    p.a=(x.a)*(y.a);
    p.b= (x.b)*(y.b);
    return p;}
```

4) Question 4 :

```
// 4) La fonction PGCD qui calcule le PGCD de deux entiers
int PGCD (int x, int y) {
    int t;
    while (y!=0) {
        t=x;
        x=y;
        y=t%y;
    }
    return x;}
```

5) Question 5 :

```
// 5) La fonction simplifier qui simplifie un rationnel
ratio simplifier (ratio x) {
    ratio y;
    y.a = x.a/PGCD(x.a,x.b);
    y.b = x.b/PGCD(x.a,x.b);
    return y;}
```

6) Question 6 :

```
main (){
    //Déclaration des variables
    ratio x,y,s,p;
    // Saisie des variables
    printf ("x = ");
    scanf("%d/%d",&(x.a),&(x.b));
    printf ("y = ");
    scanf("%d/%d",&(y.a),&(y.b));
    // Calculs et simplifications
    s=simplifier(somme(x,y));
    p=simplifier(produit(x,y));
    // Affichage des résultats
    printf ("x + y = %d/%d\n",s.a,s.b);
    printf ("x * y = %d/%d\n",p.a,p.b);
    getch();
}
```

**9-3) Exercice 5-> :****1) Question 1 :**

```
// 1) Définition des structures
// Structure date
struct date {
    int j;
    int m;
    int a;
};
typedef struct date date;
//Structure stagiaire
struct stagiaire {
    char nom[20];
    char prenom[20];
    float note;
    date dnaiss;
};
typedef struct stagiaire stg;
```

**2) Question 2 :**

```
// 2) Fonction d'affichage
void afficher (stg s){
    puts(s.nom);
    puts(s.prenom);
    printf("%.2f\n",s.note);
    printf("%d/%d/%d\n",s.dnaiss.j,s.dnaiss.m,s.dnaiss.a);}
```

**3) Question 3 :**

```
// 3) Fonction de lecture
void lire (stg *s){
    printf("nom: ");
    gets((*s).nom);
    printf("prenom: ");
    gets((*s).prenom);
    printf("note: ");
    scanf("%f",&(*s).note));
    printf("Date de naissance: ");
    scanf("%d/%d/%d\n ",&(*s).dnaiss.j,&(*s).dnaiss.m,&(*s).dnaiss.a);}
```

**4) Question 4 :**

```
// 4) Fonction qui retourne le stagiaire ayant la meilleure note
stg max (stg s[],int n){
    stg majorant;
    int i;
    majorant = s[0];
    for(i=0;i<n;i++){
        if(majorant.note < s[i].note)
            majorant = s[i];}
    return majorant;
}
```

5) Question 5 :

```
// 5) Fonction qui calcule la moyenne de la classe
float moyenne (stg s[],int n){
    int i;
    float som=0;
    for(i=0;i<n;i++){
        som=som+s[i].note;
    }
    return som/n;}
```

6) Question 6 :

```
// 6) Fonction de tri par ordre de mérite
void trimerite (stg s[],int n){
    stg t;
    int i,x=0;
    while (x==0){
        x=1;
        for(i=0;i<n-1;i++)
        {
            if(s[i].note<s[i+1].note){
                //échange des stagiaires
                t=s[i];
                s[i]=s[i+1];
                s[i+1]=t;
                x=0;
            }
        }
    }
}
```

7) Question 7 :

```
// 7) Fonction de tri par ordre alphabétique
void trialpha (stg s[],int n){
    stg t;
    int i,x=0;
    while (x==0)
    {
        x=1;
        for(i=0;i<n-1;i++){
            if(strcmp(s[i].nom,s[i+1].nom)>0)
            {
                //échange des stagiaires
                t=s[i];
                s[i]=s[i+1];
                s[i+1]=t;
                x=0;
            }
        }
    }
}
```

## 8) Question 8 :

```

//Programme principal
main () {
    stg s[3];
    int i;
    float moy;
    stg maj;
    //la saisie des stagiaires
    for(i=0;i<3;i++){
        lire(&s[i]);
        // Pour effacer le buffer
        while(getchar() !='\n');
    }
    // Marquer des retours à la ligne et séparer l'écran
    printf ("\n-----\n");
    // Calcul et affichage du nom et prénom du stagiaire ayant la meilleure
    note
    maj = max (s,3);
    printf ("le stagiaire ayant la meilleure note est : %s %s
    (%.2f)\n",maj.nom,maj.prenom,maj.note);
    // Calcul et affichage de la moyenne de la classe
    moy = moyenne (s,3);
    printf ("La moyenne de la classe est : %.2f\n",moy);
    // Affichage des stagiaires ayant une note >= à la moyenne
    printf ("La liste des stagiaires ayant une note >= %.2f :\n\n",moy);
    trimerite(s,3);
    for (i=0;i<3;i++) {
        if(s[i].note>=moy) {
            afficher(s[i]);
            printf ("\n-----\n");
        }
    }
    getch();
}

```

## 9-4) Exercice 4-9 :

## 1) Question 1 :

```

// 1) Structure heure
typedef struct heure {
    int h;
    int m;
}heure;

```

## 2) Question 2 :

```

// 2) La fonction d'affichage
void afficher (heure a){
    printf ("%d : %d\n",a.h,a.m);
}

```

3) Question 3 .

```
// 3) La fonction de saisie
void lire (heure *a){
    scanf ("%d : %d", &a->h, &a->m);
}
```

4) Question 4 :

```
// 4) La fonction vérifiant si l'heure est valide ou non
int siheure (heure a){
    if(a.h>=0 && a.h<=23 && a.m>=0 && a.m<=59) return 1;
    else return 0;
}
```

5) Question 5 :

```
// 5 La fonction qui ajoute une minute
heure plus1 (heure a){
    heure b;
    b.h=b.m=-1;
    if(a.m<=58 && a.h>=0 && a.h<=23) {
        b.m = a.m+1;
        b.h = a.h;
    }
    else if(a.m==59 && a.h>=0 && a.h<23) {
        b.m = 0;
        b.h = a.h+1;
    }
    else if(a.m==59 && a.h==23) {
        b.m = 0;
        b.h = 0;
    }
    return b;
}
```

6) Question 6 :

```
// 6) La fonction qui soustrait une minute
heure moins1 (heure a){
    heure b;
    b.h=b.m=-1;
    if(a.m>0 && a.h>=0 && a.h<=23){
        b.m = a.m-1;
        b.h = a.h;}
    else if(a.m==0 && a.h>0 && a.h<=23){
        b.m = 59;
        b.h = a.h-1; }
    else if(a.m==0 && a.h==0) {
        b.m = 59;
        b.h = 23; }
    return b;
}
```

## 7) Question 7 .

```
// 7) Fonction qui calcule la durée  
// Méthode 1:  
heure duree1 (heure a,heure b){  
    heure c;  
    c.h=c.m=-1;  
    if(b.m<a.m) {  
        b.m=b.m+60;  
        b.h=b.h-1;  
    }  
    if(b.h>=a.h) {  
        c.m=b.m-a.m;  
        c.h=b.h-a.h;  
    }  
    return c;  
}
```

## // Méthode 2:

```
heure duree2 (heure a,heure b){  
    heure c;  
    c.h=c.m=0;  
    while(a.h!=b.h || a.m!=b.m){  
        c=plus1(c);  
        b=moins1(b);  
    }  
    return c;  
}
```

## 8) Question 8 :

```
// 8) Programme principal  
main (){  
    heure h1,h2,d;  
    do {  
        printf ("saisir l'\heure de depart : ");  
        lire (&h1);  
    } while (siheure(h1)==0);  
    do {  
        printf ("saisir l'\heure d'\arrivee : ");  
        lire (&h2);  
    } while (siheure(h2)==0);  
    d=duree2(h1,h2);  
    if (d.h!=-1) {  
        printf ("La duree du voyage : ");  
        afficher(d);  
    }  
    else printf ("\nErreur :\nHeure de depart > Heure d'\arrivee");  
  
    getch();  
}
```

**9-5) Exercice 5-> :**

1) Question 1 :

```
#include <stdio.h>
#include <conio.h>
// 1) La structure date
typedef struct date {
    int j;
    int m;
    int a;
}date;
```

2) Question 1 :

```
// 2) La fonction d'affichage
void afficher (date d){
    printf ("%d/%d/%d",d.j,d.m,d.a);
}
```

3) Question 1 :

```
// 3) La fonction de lecture
void lire (date *d){
    scanf ("%d/%d/%d",&d->j,&d->m,&d->a);}
```

4) Question 1 :

```
// 4) La fonction qui vérifie si une année est bissextile ou non
int sibissextile (int an){
    if(an%400==0 || an%4==0 && an%100!=0) return 1;
    else return 0;}
```

5) Question 1 :

```
// 5) La fonction ajoute un jour
date plus1 (date d){
    int valide = 1;
    date x;
    x.j=x.m=x.a=0;
    //Les mois de 30 jours
    if(d.m==4 || d.m==6 || d.m==9 || d.m==11){
        if(d.j>1 && d.j<30) d.j=d.j+1;
        else if (d.j==30) {
            d.j=1;
            d.m=d.m+1;
        }
        else valide = 0;}
    //Les mois de 31 jours sauf le mois 12
    else if (d.m==1 || d.m==3 || d.m==5 || d.m==7 || d.m==8 || d.m==10){
        if(d.j>1 && d.j<31) d.j=d.j+1;
        else if(d.j==31) {
            d.j=1;
            d.m=d.m+1;
        }
        else valide = 0;}}
```

```

//Le mois 12
else if (d.m==12){
    if(d.j>=1 && d.j<31) d.j=d.j+1;
    else if(d.j==31) {
        d.j=d.m=1;
        d.a=d.a+1;
    }
    else valide = 0;}
//Le mois 2
else if(d.m==2){
    //Année bissextile
    if(d.a%400==0 || d.a%4==0 && d.a%100!=0){
        if(d.j>=1 && d.j<29) d.j=d.j+1;
        else if (d.j==29) {
            d.j=1;
            d.m=d.m+1;
        }
        else valide = 0;}
    //Année non bissextile
    else {
        if(d.j>=1 && d.j<28) d.j=d.j+1;
        else if (d.j==28) {
            d.j=1;
            d.m=d.m+1;
        }
        else valide = 0;
    }
}
//Le reste des valeurs des mois
else valide = 0;
if(valide != 0) return d;
else return x;

```

6) Question 1 :

// 6) La fonction moins1 qui soustrait un jour

```

date moins1 (date d){

    int valide = 1;
    date x;
    x.j=x.m=x.a=0;

    //Les mois de 30 jours et les mois avant de 31 jours
    if(d.m==4 || d.m==6 || d.m==9 || d.m==11){
        if(d.j>1 && d.j<=30) d.j=d.j-1;
        else if (d.j==1) {
            d.j=31;
            d.m=d.m-1;
        }
        else valide = 0; }

```

```
//Les mois de 31 jours et les mois avant de 31 jours
else if(d.m==5 || d.m==7 || d.m==10 || d.m==12)
{
    if(d.j>1 && d.j<=31) d.j=d.j-1;
    else if (d.j==1) {
        d.j=30;
        d.m=d.m-1;
    }
    else valide = 0;
}

//Les mois de 31 jours et les mois avant de 31 jours
else if(d.m==8){
    if(d.j>1 && d.j<=31) d.j=d.j-1;
    else if (d.j==1) {
        d.j=31;
        d.m=d.m-1;
    }
    else valide = 0;
}
//Le mois 1
else if(d.m==1){
    if(d.j>1 && d.j<=31) d.j=d.j-1;
    else if (d.j==1) {
        d.j=31;
        d.m=12;
        d.a=d.a-1;
    }
    else valide = 0;}
//Le mois 2
else if(d.m==2){
    if(d.a%400==0 || d.a%4==0 && d.a%100!=0){
        if(d.j>1 && d.j<=29) d.j=d.j-1;
        else if (d.j==1) {
            d.j=31;
            d.m=d.m-1;
        }
        else valide = 0;
    }
    else {
        if(d.j>1 && d.j<=28) d.j=d.j-1;
        else if (d.j==1) {
            d.j=31;
            d.m=d.m-1;
        }
        else valide = 0;
    }
}
```

```
//Le mois 3
else if(d.m==3){
    if(d.j>1 && d.j<=31) d.j=d.j-1;
    else if (d.j==1) {
        if(d.a%400==0 || d.a%4==0 && d.a%100!=0) d.j=29;
        else d.j=28;
        d.m=d.m-1;
    }
    else valide = 0;
}
if(valide != 0) return d;
else return x;}
```

7) Question 1 :

```
// 7) la fonction qui calcule le nombre de jours entre deux dates
int duree (date d1,date d2){
    int c=0;
    while(d1.j!=d2.j || d1.m!=d2.m || d1.a!=d2.a){
        c++;
        d2=moins1(d2);}
    return c;}
```

8) Question 1 :

```
// 8) Programme principal
main (){
    date da,dn;
    int age;
    printf ("Saisir votre date de naissance (JJ/MM/AA) : ");
    lire (&da);
    printf ("Saisir la date actuelle (JJ/MM/AA) : ");
    lire (&dn);
    age = duree(da,dn);
    printf ("Votre age en jours est : %d jours",age);
    getch(); }
```

### 9-6) Exercice 6-9 :

1) Question 1 :

```
// 1) La structure ip (4 octets=4 entiers)
typedef struct ip {
    short o1;
    short o2;
    short o3;
    short o4;
}ip;
```

2) Question 1 :

```
// 2) La fonction d'affichage showip
void showip (ip a){
    printf ("%d.%d.%d.%d\n",a.o1,a.o2,a.o3,a.o4);}
```

3) Question 1 .

```
// 3) La fonction de lecture readip
void readip (ip *a){
    scanf ("%d.%d.%d.%d",&a->o1,&a->o2,&a->o3,&a->o4);
}
```

4) Question 1 :

```
// 4) La fonction qui vérifie la validité d'une adresse ip
int ifip (ip a){
    if(a.o1>=0 && a.o1<=255 && a.o2>=0 && a.o2<=255
        && a.o3>=0 && a.o3<=255 && a.o4>=0 && a.o4<=255)
        return 1;
    else return 0;
}
```

5) Question 1 :

```
// 5) La fonction qui vérifie la validité d'un masque
int ifmask (int mask){
    if (mask<=0 || mask>=31) return 0;
    else return 1;
}
```

6) Question 1 :

```
// 6) La fonction qui retourne la classe d'une adresse
char classe (ip a){
    if (a.o1>=0 && a.o1<=127) return 'A';
    else if (a.o1>=128 && a.o1<=191) return 'B';
    else if (a.o1>=192 && a.o1<=223) return 'C';
    else if (a.o1>=224 && a.o1<=239) return 'D';
    else if (a.o1>=240 && a.o1<=255) return 'E';}
```

7) Question 1 :

```
// 7) La fonction qui convertit le masque en écriture décimale
ip maskdec(int mask)
{
    ip a;
    if (mask<=8) {
        a.o1=256-pow(2,8-mask);
        a.o2=0;
        a.o3=0;
        a.o4=0;
    }
    else if (mask>8 && mask<=16) {
        a.o1=255;
        a.o2=256-pow(2,16-mask);
        a.o3=0;
        a.o4=0;
    }
    else if (mask>16 && mask<=24) {
        a.o1=255;
        a.o2=255;
        a.o3=256-pow(2,24-mask);
        a.o4=0;
    }
}
```

```

        else if (mask>2<sup>32 - mask</sup>)
    {
        a.o1=255;
        a.o2=255;
        a.o3=255;
        a.o4=256-pow(2,32-mask);
    }

    return a;
}

```

8) Question 1 :

```

// 8) La fonction qui calcule l'adresse réseau
ip netip (ip a, int mask)
{
    ip MASK; // Décimal
    MASK=maskdec(mask);
    a.o4=a.o4 & MASK.o4;
    a.o3=a.o3 & MASK.o3;
    a.o2=a.o2 & MASK.o2;
    a.o1=a.o1 & MASK.o1;
    return a;
}

```

9) Question 1 :

```

// 9) La fonction qui calcule l'adresse de diffusion
ip broadip (ip a, int mask){
    ip MASK; // Décimal
    MASK=maskdec(mask);
    a.o4=(a.o4 | ~(MASK.o4))&255;
    a.o3=(a.o3 | ~(MASK.o3))&255;
    a.o2=(a.o2 | ~(MASK.o2))&255;
    a.o1=(a.o1 | ~(MASK.o1))&255;
    return a;
}

```

10) Question 1 :

```

// 10) La fonction qui calcule la première adresse
ip firstip (ip a, int mask){
    ip b;
    b=netip(a,mask);
    b.o4= b.o4 + 1;
    return b;
}

```

11) Question 1 :

```

// 11) La fonction qui calcule la dernière adresse
ip lastip (ip a,int mask){
    ip b;
    b=broadip(a,mask);
    b.o4= b.o4 - 1;
    return b;
}

```

12) Question 1 .

```
main (){
    ip a,net_a,broad_a,first_a,last_a;
    int m;
    printf ("saisir une adresse IP: ");
    readip (&a);
    printf ("saisir le masque : ");
    scanf ("%d",&m);
    net_a = netip(a,m);
    broad_a = broadip (a,m);
    first_a = firstip (a,m);
    last_a = lastip (a,m);
    printf ("\nL\'adresse reseau : ");
    showip(net_a);
    printf ("La premiere adresse : ");
    showip(first_a);
    printf ("La derniere adresse : ");
    showip(last_a);
    printf ("L\'adresse de diffusion : ");
    showip(broad_a);
    getch();
}
```

## 10. Les fichiers

### 10-1) Exercice 1-10 :

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main()
{
    FILE *f;
    char nom[20];
    printf ("Saisir votre nom: ");
    gets(nom);
    f=fopen("nom.txt","w");
    fputs(nom,f);
    fclose(f);
    getch();
}
```

### 10-2) Exercice 2-10 :

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

void AddToFile(char FileName[20],char Nom[20])
{
    FILE *f;
    f=fopen(FileName,"a+");
    fputs(Nom,f);
    fputs("\n",f);
    fclose(f);
}

void ReadFile(char FileName[20],char Nom[20]){
    FILE *f;
    f=fopen(FileName,"r");
    fread(Nom,sizeof(char),100,f);
    fclose(f);
}

int main() {
    FILE *f;
    char nom[20];
    char noms[100];
    char filename[20];
    printf ("Saisir le nom du fichier : ");
    gets(filename);
    printf ("Saisir votre nom: ");
```

```
    gets(nom);
    AddToFile(filename,nom);
    ReadFile(filename,noms);
    printf ("Le contenu du fichier %s est :\n",filename);
    puts(noms);
    getch();
}
```

### 10-3) Exercice 3-10 :

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
void AddToFile(char FileName[20],char Nom[20])
{
    FILE *f;
    f=fopen(FileName,"a+");
    fputs(Nom,f);
    fputs("\n",f);
    fclose(f);
}
void ReadFile(char FileName[20],char Nom[20])
{
    FILE *f;
    f=fopen(FileName,"r");
    fread(Nom,sizeof(char),100,f);
    fclose(f);
}
int main()
{
    FILE *f;
    char nom[20];
    char noms[100];
    char filename[20];
    printf ("Saisir le nom du fichier : ");
    gets(filename);
    do{
        printf ("Saisir un nom: ");
        gets(nom);
        if(strcmp(nom,"")!=0)
            AddToFile(filename,nom);
    } while (strcmp(nom,"")!=0);
    ReadFile(filename,noms);
    printf ("Le contenu du fichier %s est :\n",filename);
    puts(noms);
    getch();
}
```

**10-4) Exercice 4-10 :**

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int LengthFile(char FileName[20]) {
    FILE *f;
    int len;
    f=fopen(FileName,"r");
    fseek(f,0,SEEK_END);
    len=f.tell(f);
    fclose(f);
    return len;
}
int main()
{
    int l;
    l=LengthFile("noms.txt");
    printf("La longueur de la chaine est : %d octets", l);
    getch();
}
```

**10-5) Exercice 5-10 :**

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int SearchChar(char FileName[20], char c) {
    FILE *f;
    char ch[100];
    int i,nbr=0;
    f=fopen(FileName,"r");
    fread(ch,sizeof(char),100,f);
    fclose(f);
    for(i=0;i<strlen(ch);i++) {
        if(ch[i]==c)
            nbr++;
    }
    return nbr;
}
int main(){
    int l;
    char c;
    c=getchar();
    l=SearchChar("noms.txt",c);
    printf("Le nombre d'occurrences de %c est : %d",c,l);
    getch();}
```

## 10-6) Exercice 9-10 :

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int SearchStr(char FileName[20], char c[20])
{
    FILE *f;
    char ch[100];
    char *t;
    int nbr=0;
    f=fopen(FileName,"r");
    fread(ch,sizeof(char),100,f);
    fclose(f);
    puts(ch);
    t=strstr(ch,c);

    while(t!=NULL)
    {
        nbr++;
        t=strstr(t+strlen(c),c);
    }
    return nbr;
}

int main()
{
    int l;
    char c[20];
    printf ("Saisir une chaine :");
    gets(c);
    l=SearchStr("noms.txt",c);
    printf("Le nombre d'occurences de %s est : %d",c,l);
    getch();
}
```

## 10-7) Exercice 7-10 :

```
#include <stdio.h>
#include <string.h>
#include <conio.h>

int NbrLine(char FileName[20]){
    FILE *f;
    char ch[100];
    int i,nbr=0;
    f=fopen(FileName,"r");
```

```

        fread(ch,sizeof(char),100,f),
        fclose(f);
        for(i=0;i<strlen(ch);i++) {
            if(ch[i]=='\n')
                nbr++;
        }
        return nbr+1;
    }
int main()
{
    int l;
    char filename [20];
    printf ("Saisir le nom du fichier : ");
    gets(filename);
    l=NbrLine(filename);
    printf("Le nombre de lignes est : %d",l);
    getch();
}

```

### 10-8) Exercice 8-10 :

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
int FirstChar(char FileName[20], char c[20])
{
    FILE *f;
    char ch[100];
    char *x, *t;
    t=calloc(sizeof(char),20);
    t[0] = ' ';
    int nbr=0;
    f=fopen(FileName,"r");
    fread(ch,sizeof(char),100,f);
    fclose(f);
    puts(ch);
    strcat(t,c);puts(t);
    x=strstr(ch,t);
    while(x!=NULL)
    {
        nbr++;
        x=strstr(x+2,t);
    }
    return nbr;
}

```

```
int main()
{
    int l;
    char c[20];
    printf ("Saisir une chaîne :");
    gets(c);
    l=FirstChar("noms.txt",c);
    printf("Le nombre d'occurrences de %s est : %d",c,l);
    getch();
}
```

### 10-9) Exercice 9-10 :

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
void ReplaceChar(char FileName[20], char *c,char *d)
{
    FILE *f;
    char *e;
    int len = 0;
    e=malloc(sizeof(char),1);
    char *ch;
    ch=malloc(sizeof(char),100);
    int i;

    /* Ouverture du fichier en mode lecture
       (le fichier doit exister.Sinon utiliser le mode r+) */
    f=fopen(FileName,"r");
    // Lecture du fichier caractère par caractère
    while(!feof(f)){
        fread(e,sizeof(char),1,f);
        if(!feof(f))strcat(ch,e);
        len++;
    }
    for(i=0;i<len;i++)
    {
        if(ch[i]==*c)
            ch[i]=*d;
    }
    fclose(f);

    f=fopen(FileName,"w+");
    fwrite(ch,sizeof(char),len-1,f);
    fclose(f);
}
```

```

int main(){
    char c,d;
    char filename[20];
    printf("Saisir le nom du fichier : ");
    gets(filename);
    fflush(stdin);
    printf("Saisir le caractère à remplacer : ");
    c=getchar();
    fflush(stdin);
    printf("Saisir le caractère remplaçant : ");
    d=getchar();
    ReplaceChar("noms.txt",&c,&d);
    getch();
}

```

### 10-10) Exercice 10-10 :

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
void ReplaceStr(char FileName[20], char c[20],char d[20])
{
    FILE *f;
    char *e, *t;
    char *res;
    int len = 0;
    int indice ;
    int l = strlen(d)-strlen(c);
    e=malloc(sizeof(char),20);
    char *ch;
    ch=calloc(sizeof(char),100);
    int i;
    f=fopen(FileName,"r");
    while(!feof(f)){
        fread(e,sizeof(char),1,f);
        if(!feof(f))strcat(ch,e);
        len++;
    }
    t=strstr(ch,c);
    while(t!=NULL){
        indice = strlen(ch)-strlen(t);
        // Déplacer à droite
        if (l>0)
        {
            for (i=strlen(ch)-1;i>=indice+strlen(c);i--) ch[i+1]= ch[i];
            strncpy(t,d,strlen(d));
        }
    }
}

```

```

//Déplacer à gauche
else if(l<0) {
    strncpy(t,d,strlen(d));
    for (i=indice+strlen(d);i<strlen(ch);i++) ch[i]= ch[i-1];
}
t=strstr(t+strlen(d),c);
len = len + 1;
}
fclose(f);
f=fopen(FileName,"w+");
fwrite(ch,sizeof(char),len-1,f);
fclose(f);
}

int main()
{
    char c[20],d[20];
    char filename[20];
    printf("Saisir le nom du fichier : ");
    gets(filename);
    fflush(stdin);
    printf("Saisir la chaîne à remplacer : ");
    gets(c);
    fflush(stdin);
    printf("Saisir la chaîne remplacante : ");
    gets(d);
    ReplaceStr(filename,c,d);
    getch();
}

```

### 10-11) Exercice 11-10 :

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
void DeleteChar(char FileName[20], char c){
    FILE *f;
    char *t, *e;
    int len = 0;
    char *ch;
    ch=calloc(sizeof(char),100);
    e=calloc(sizeof(char),1);
    int i,indice;
    f=fopen(FileName,"r");
    while(!feof(f)){
        fread(e,sizeof(char),1,f);
        if(!feof(f))strcat(ch,e);
        len++;
    }
    printf ("Avant : \n");

```

```

puts(ch);
t=strchr(ch,c);
while(t!=NULL){
    indice = strlen(ch) - strlen (t);
    for (i=indice;i<strlen(ch);i++) ch[i]= ch[i+1];
    t=strchr(t+1,c);
    len = len - 1;
}
printf ("\nApres : \n");
puts(ch);
fclose(f);
f=fopen(FileName,"w+");
fwrite(ch,sizeof(char),len-1,f);
fclose(f);
}
int main()
{
    char c;
    char filename[20];
    printf("Saisir le nom du fichier : ");
    gets(filename);
    fflush(stdin);
    printf("Saisir le caractere a supprimer : ");
    c=getchar();
    DeleteChar("noms.txt",c);
    getch();}
```

### 10-12) Exercice 12-10 :

```

#include <stdio.h>
#include <string.h>
#include <conio.h>
void DeleteStr(char FileName[20], char c[20])
{
    FILE *f;
    char *t, *e;
    int len = 0;
    char *ch;
    ch=calloc(sizeof(char),100);
    e=calloc(sizeof(char),1);
    int i,indice;
    f=fopen(FileName,"r");
    while(!feof(f)){
        fread(e,sizeof(char),1,f);
        if(!feof(f))strcat(ch,e);
        len++;
    }
    printf ("Avant : \n");
    puts(ch);
```