

Exercice N°1 : Nombre d'occurrence

Ecrire un programme qui permet de remplir une matrice d'ordre **N** (**N** compris entre **2** et **10**) au hasard par des entiers de trois chiffres, calculer et d'afficher le nombre d'occurrences d'un chiffres **C** donné.

Exemple :

Pour **N = 4** et la matrice carré **M** ci-dessous :

C = 5

	0	1	2	3
0	124	809	509	423
1	587	250	102	586
2	506	100	390	145
3	589	608	940	358

Le programme affichera : Le nombre d'occurrence de **5** est **8**

Exercice N°2 : Composition

Ecrire un programme qui permet :

- Saisir le nombre de lignes et de colonnes **N** (**N** compris entre **2** et **10**)
- Remplir une matrice carrée au hasard par des lettres majuscules.
- Afficher les chaines composées par les lettres de chaque ligne et chaque colonne.
- Afficher la chaine composée par les lettres de la diagonale et l'antidiagonale.

Exemple :

Pour **N=4** et la matrice carrée **M** suivante :

	0	1	2	3
0	A	L	I	G
1	R	Y	U	M
2	O	F	P	E
3	S	N	R	Q

Le programme affichera :

Les chaines des lignes : "ALIG", "RYUM", "OFPE", "SNRQ"

Les chaines des colonnes : "AROS", "LYFN", "IUPR", "GMEQ"

La chaine de la diagonale : "AYPQ"

La chaine de l'anti-diagonale : "GUFS"

Exercice N°3 : Matrice d'identité

La matrice identité d'ordre **N** est une matrice carré dont les valeurs des cases de la première diagonale sont égales à 1 alors que celle se trouvant dans les autres cases sont nulles.

Ecrire un programme qui permet de vérifier si une matrice d'entiers d'ordre **N** (**N** compris entre 2 et 10) est une matrice identité

Exemple :

Pour **N=5** et la matrice carrée **M** ci-dessous :

Le programme affichera :

M est une matrice d'identité de taille **5**

	0	1	2	3	4
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1

Exercice N°4 : Transposée d'une matrice

Ecrire un programme qui permet :

- Saisir le nombre de lignes (L compris entre 2 et 10)
- Saisir le nombre de colonnes (C compris entre 2 et 10)
- Remplir une matrice **M** par des entiers.
- Faire la transposition de la matrice saisie en une matrice **MT**.

Exemple :

Saisir le nombre de ligne : 2

M

	0	1	2	4
0	1	2	3	4
1	5	6	7	8

MT

	0	1
0	1	5
1	2	6
2	3	7
3	4	8

Exercice N°5 : Matrice symétrique

Une matrice symétrique est une matrice carré d'ordre **N** qui est égale à sa propre transposée, ainsi **M** est symétrique si transposé **MT** = **M**

Exemple :Pour **N = 3** et la matrice carrée **M** ci-dessous :

	0	1	2
0	0	1	-2
1	-1	0	3
2	2	-3	0

	0	1	z
0	0	1	-2
1	-1	0	3
2	2	-3	0

Ecrire un programme qui permet de vérifier si une matrice d'entiers d'ordre **N** (**N** compris entre 2 et 10) est une matrice symétrique.

Exercice N°6 : Transfert

Ecrire un programme qui permet de transférer les éléments d'une matrice **M** d'entiers d'ordre **N** (**N** compris entre 2 et 10) dans tableau **T** d'entiers.

Exemple :Pour **N = 3** et la matrice carrée **M** ci-dessous :

	0	1	2
0	2	4	6
1	4	0	10
2	6	10	12

devient

t	2	4	6	4	0	10	6	10	12
0	2	4	6	4	0	10	6	10	12
1	4	0	10	4	0	10	6	10	12
2	6	10	12	4	0	10	6	10	12

Exercice N°7 : Matrice stochastique ou bistochastique

En mathématique, une matrice **stochastique** (aussi appelée matrice de Markov) est une matrice carrée dont chaque élément est **un réel** compris entre 0 et 1 et dont **la somme des éléments de chaque ligne vaut 1**.

Une matrice est dite **bistochastique** (ou doublement stochastique) si **la somme des éléments de chaque ligne et de chaque colonne vaut 1**.

	0	1	2
0	0.5	0.3	0.2
1	0.2	0.8	0
2	0.3	0.3	0.4

Le programme affichera : M est une matrice stochastique

Écrire un programme qui permet de vérifier si une matrice **M** d'entiers d'ordre **N** (**N** compris entre 2 et 10) est une matrice stochastique ou bistochastique

Exercice N°8 : Concaténation

On désire former, à partir d'une matrice **M** de **L lignes * C colonnes** d'entiers formés par un seul chiffre, un tableau **T** de **L éléments**, dont chaque élément de **T** est l'entier formé par la concaténation des éléments de chaque ligne de la matrice **M**.

Exemple :

Pour la matrice **M** de **3 lignes et 4 colonnes**.

	0	1	2	3
0	7	4	3	0
1	1	5	2	6
2	6	9	1	9

On aura le tableau **T** suivant :

T	7430	1526	6919
	0	1	2

Écrire un programme qui permet :

- Saisir le nombre de lignes (**L** compris entre 2 et 10).
- Saisir le nombre de colonnes (**C** compris entre 2 et 10).
- Remplir la matrice **M** par des entiers formés par un seul chiffre.
- Former le tableau **T** comme définie ci-dessus.

Exercice N°9 : Points-cols

On appelle **point-col** dans une matrice, les éléments qui **sont à la fois un maximum sur leur ligne et un minimum sur leur colonne**.

Écrire un programme qui permet d'afficher les coordonnées des points-cols dans une matrice **M** d'entiers d'ordre **N** (**N** compris entre 2 et 10).

Exemple :

Pour **N = 3** et la matrice carrée **M** suivante :

	0	1	2
0	4	9	6
1	1	12	3
2	7	18	9

Le programme affichera : M [0,1] est un point-col

Exercice N°10 : Colonne dominante

Une colonne dominante est la colonne qui contient le plus des maximums des **N** lignes.

Écrire un programme qui permet d'afficher le numéro de la colonne dominante dans une matrice d'entiers d'ordre **N** (**N** compris entre 2 et 10)

Exemple :

Pour **N = 5** et la matrice carrée **M** suivante :

	0	1	2	3	4
0	12	5	18	11	9
1	21	25	5	17	19
2	31	12	29	17	14
3	13	18	16	21	48
4	7	2	43	25	22

Le programme affichera : La colonne 2 est une colonne **4ème**

Exercice N°11 : Recherche

Écrire un programme qui permet de chercher et afficher tous les entiers formées de trois chiffres d'une matrice M d'entiers d'ordre N (N compris entre 2 et 10) qui donnent la même valeur si on le lit de gauche à droite ou de droite à gauche.

Exemple :

Pour N = 6 et la matrice carrée M suivante :

	0	1	2	3	4	5
0	123	245	100	121	205	100
1	122	180	306	140	586	123
2	222	122	454	239	650	351
3	301	897	105	798	602	152
4	750	453	465	289	476	303
5	800	375	149	415	913	102

Le programme affichera : 121 222 454

303

Exercice 12 : Cryptage

Une manière de coder un texte consiste en l'opération suivante.

On considère que le texte à coder est une chaîne de caractères.

En commençant à partir du premier caractère, on divise le texte en blocs de longueur 3 caractères consécutifs.

Si la longueur du texte n'est pas un multiple de 3, on lui ajoute autant d'espaces qu'il en faut pour que sa longueur devienne un multiple de 3.

En disposant ces blocs l'un en dessous de l'autre, on obtient alors 3 colonnes.

Chaque colonne, lue du haut vers le bas, va donner une chaîne de caractères.

En rassemblant dans l'ordre toutes ces chaînes, on obtient une nouvelle chaîne qui sera le texte codé.

Exemple :

Si le texte à coder = "J'aime la programmation"

"J"	" "	"a"	"i"	"m"	"e"	" "	"l"	"a"	" "	"p"	"r"	"o"	"g"	"r"		"m"	"m"	"a"	"t"	"i"	"o"	"n"
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

La longueur de ce texte est égale à 23.

On lui ajoute **1 espace** pour que sa longueur devienne multiple de 3.

La chaîne devient "J'aime la programmation "

Espace

Sa longueur devient donc **24**.

	0	1	2
0	J	'	a
1	i	m	e
2		l	a
3		p	r
4	o	g	r
5	a	m	m
6	a	t	i
7	o	n	

En appliquant les étapes ci-dessus évoquées, on obtient :

Le texte codé sera : "**Ji oao'mlpgmtnaearrmi**"

Écrire un programme qui permet :

- Saisir un texte contenant au moins 4 caractères.
- Coder et afficher le texte en respectant la méthode de cryptage citée ci-dessus.

Exercice 13: Tri d'une matrice

Écrire un programme qui permet :

- Remplir une matrice d'entier d'ordre N (N compris entre 2 et 10) au hasard par des entiers non nuls et inférieurs à 100.
- Trier la matrice M selon principe suivant :

- La première en ordre croissant
- La deuxième ligne en ordre décroissant
- La troisième ligne en ordre croissant.
-
- Afficher la matrice **M** triée.

Exemple :

Pour **N = 3** et la matrice carré **M** ci-dessous :

	0	1	2
0	9	2	5
1	1	7	6
2	4	10	3

devient **M**

	0	1	2
0	2	5	9
1	7	6	1
2	3	4	10

Exercice 14: Bac Théorique 2023 [durée= 30 min, 5 points]

Une image en niveaux de gris est un ensemble de pixels structurés sous forme d'une matrice **M1** de **L1** lignes et **C1** colonnes ou chaque case contient un code hexadécimal formé de 2 caractères.

En appliquant la méthode de réPLICATION de pixels pour zoomer (agrandir) **K** fois cette image (avec **K** est un entier strictement positif donné), on obtient une matrice **M2** formé de **L1 * K** lignes et **C1*K** colonnes ou chaque pixel de la matrice **M1** est répliqué (recopié) **K** fois lignes et **K** fois colonnes.

Exemple :

Pour la matrice **M1** de **3** lignes et **4** colonnes ci-dessous contenant les pixels de l'image à zoomer et **k=2**

	0	1	2	3
M1	BD	72	BB	E5
	DD	9F	E9	20
	86	4C	38	B6

La matrice **M2** contenant l'image agrandir aura la taille $(3*K)*(4*L)$ c'est-à-dire **6*8** et aura le contenu ci-après où chaque élément de la matrice **M1** est réplié **K*K** fois ($2*2=4$ fois)

	0	1	2	3	4	5	6	7
M2	BD	BD	72	72	BB	BB	E5	E5
	BD	BD	72	72	BB	BB	E5	E5
	DD	DD	9F	9F	E9	E9	20	20
	DD	DD	9F	9F	E9	E9	20	20
	86	86	4C	4C	38	38	B6	B6
	86	86	4C	4C	38	38	B6	B6

On se propose de remplir un fichier texte F par les données relatives à l'image agrandie comme suit :

- La première ligne contiendra le nombre de lignes de la matrice **M1**.
- La deuxième ligne contiendra le nombre de colonnes de la matrice **M1**.
- La troisième ligne contiendra l'échelle de zoom **K**.
- Chacune des autres lignes contiendra la concaténation des valeurs contenues dans une ligne de la matrice **M2** tout en les séparant par le caractère "|".

Travail demandé :

- 1) Ecrire l'instruction d'ouverture de fichier **F** sachant que ce fichier est stocké dans la racine du disque **D** sous le nom "**Image.txt**"

- 2) En utilisant une procédure **Remplissage(M1, M2, L1, C1, K)** qui permet de remplir la matrice **M2**, écrire un algorithme d'un module **Remplir(M1, L1,C1,K,F)** qui permet de remplir le fichier F par les données relatives à l'image agrandie comme décrit précédemment (avec **M1** est type **Mat**)

N.B: Le candidat n'est pas appelé à développer la procédure REMPLISSAGE ni à réaliser les tâches suivantes :

- La saisie de **L1, C1 et K**.
- Le remplissage de **M1** et le définition de son type **Mat**.

Exercice 15: Bac Théorique 2021 [durée= 30 min, 4.75 points]

Les points cols d'une matrice **M** d'entiers sont les éléments qui sont minimum sur leur ligne et maximum sur leur colonne.

Exemple : Pour la matrice **M** suivante :

	0	1	2	3
0	9	5	7	5
1	4	2	2	3
2	4	3	3	2
3	8	5	6	5
4	7	4	10	4

Le points cols sont : **M[0,1], M[0,3], M[3,1] et M[3,3]**

En effet,

M[0,1] = 5 est le minimum de la 1^{ère} ligne et le maximum de la 2^{ème} colonne.

M[0,3] = 5 est le minimum de la 1^{ère} ligne et le maximum de la 4^{ème} colonne.

M[3,1] = 5 est le minimum de la 4^{ème} ligne et le maximum de la 2^{ème} colonne.

M[3,3] = 5 est le minimum de la 4^{ème} ligne et le maximum de la 4^{ème} colonne.

On présente ci-après une méthode pour déterminer les points cols d'une matrice **M** de **L × C** entiers :

- 1) Remplir deux matrice **M_min** et **M_max** de mêmes dimensions que **M** de sorte que :

- **M_min** contiendra des uns (1) aux positions des minimums des lignes dans **M** et des zéros dans le reste des cases.
- **M_max** contiendra des uns (1) aux positions des maximums des colonnes dans **M** et des zéros dans les reste des cases.

- 2) Remplir un fichier texte **F_col** par les valeurs de **M** dont les cases de même coordonnées, dans les deux matrices **M_min** et **M_max**, contiennent à la fois la valeur 1. Chaque ligne du fichier contiendra la valeur du point col et le couple de ses coordonnées dans la matrice **M** séparés par un espace.

Exemple : Pour la matrice **M** de l'exemple précédent,

les matrices **M_min** et **M_max** seront :

	0	1	2	3
0	0	1	0	1
1	0	1	1	0
2	0	0	0	1
3	0	1	0	1
4	0	1	0	1

Le contenu du fichier **F_col** sera :

En effet :

- Les cases **M_min[0,1]** et **M_max[0,1]** contiennent à la fois la valeur 1
- Les cases **M_min[0,3]** et **M_max[0,3]** contiennent à la fois la valeur 1
- Les cases **M_min[3,1]** et **M_max[3,1]** contiennent à la fois la valeur 1
- Les cases **M_min[3,3]** et **M_max[3,3]** contiennent à la fois la valeur 1

Travail demandé :

En utilisant le principe décrit précédemment, écrire un algorithme d'une procédure **Points_cols(M,L,C,F_col)** qui permet de remplir un fichier texte **F_col** par le(s) point(s) col(s) d'une matrice **M** de **L × C** entiers s'ils existent, sachant que **M**, **L** et **C** sont déjà saisis dans le programme appelant et **M** est de type **Mat**.

Exercice 16: Bac Théorique 2020 [durée= 30 min, 5 points]

Soit **M** une matrice carrée de $N \times N$ entiers distincts avec $N \in [5,50]$, chaque ligne contient un seul maximum. On désigne par colonne dominante d'une matrice , la colonne qui contient le plus de maximums des lignes de cette matrice.

	0	1	2	3	4
0	32	12	10	89	15
1	3	33	14	1	18
2	54	5	22	76	50
3	34	21	6	29	17
4	19	9	11	84	25

Exemple :

Pour $N=5$ et la matrice **M** ci-contre, la 4^{ème} colonne est la colonne dominante puisque parmi les 5 maximums des 5 lignes il y a 3 maximums qui se trouve dans la 4^{ème} colonne.

On se propose d'écrire un module nommé **Remplir** qui permet de générer, sur la racine du disque **D**, un fichier d'enregistrements "**F_Max.dat**" à partir de la matrice **M**, le numéro **NL** de la ligne et le numéro **NC** de la colonne de **Vmax**.

Exemple :

Pour la matrice précédente, le premier enregistrement du fichier "**F_Max.dat**" contiendra les valeurs suivantes :

Vmax = 98, NL = 0 et NC = 3

Travail demandé :

- 1) Dresser un tableau de déclaration des nouveaux types pour déclarer :
 - Un type pour le fichier "**F_Max.dat**" ainsi que tous les nouveaux types nécessaires à sa déclaration.
 - Un type pour la matrice **M**.
- 2) Donner une instruction d'association du fichier "**F_Max.dat**" à une variable logique **F**.
- 3) Développer le module **Remplir**, sachant que **M** et **N** sont déjà saisis au niveau du programme appelant.
- 4) En utilisant une fonction **Frequence(F , NC)** qui retourne le nombre d'occurrences d'un numéro de colonne **NC** dans le fichier d'enregistrement **F**, écrire un algorithme d'un module qui permet de chercher le numéro de la colonne dominante à partir du fichier "**F_Max.dat**".

N.B :

-Le candidat n'est pas appelé à développer la fonction **Frequence**.

-Lorsque plus qu'une colonne est dominante, on affichera la dernière rencontrée dans le fichier.

Exercice 17: Bac Théorique 2019 [durée =1H : 15 , 10 points]

Parmi les méthodes de chiffrement utilisant un mot-clé, on cite celle décrite ci-après qui permet de crypter un message **msg** ne dépassant pas **18** caractères et formé uniquement de lettres minuscules, de chiffres et d'espace :

Etape1 : Remplir aléatoirement une matrice carrée **M1** de dimension 6×6 par toutes les lettres alphabétiques minuscules ainsi que tous les chiffres.

NB : Les indices des lignes et des colonnes de la matrice **M1** sont les lettres **A, B, C, D, E** et **F**.

Etape2 : Générer un message intermédiaire **msgi**, en concaténant les résultats du chiffrement de chaque caractère du message **msg**. Le résultat du chiffrement d'un caractère est la concaténation de l'indice de la ligne avec l'indice de la colonne de la case contenant le caractère à chiffrer.

Le caractère espace ne sera pas chiffré.

Etape3 : Remplir une deuxième matrice **M2** de taille 7×6 caractères en mettant dans :

- La première ligne, les lettres d'un mot-clé formé de **6** lettres majuscules.
- Le reste des lignes, le message **msgi** caractère par caractère en commençant par la première case de la deuxième ligne.

NB : Chaque case vide de la matrice **M2** sera remplie par le caractère espace.

Etape4 : Trier les éléments de la 1^{ère} ligne de **M2** selon un ordre alphabétique croissant, sachant que tout déplacement d'un élément entame le déplacement de tous les éléments de la colonne correspondante.

Etape5 : Concaténer les lettres de la matrice **M2**, colonne par colonne en commençant par la 1^{ère} colonne et sans considérer les éléments de la 1^{ère} ligne, pour obtenir le message chiffré final.

	A	B	C	D	E	F
M1	c	1	o	f	w	j
	y	m	t	5	b	4
	i	7	a	2	8	s
	p	3	0	q	h	x
	k	e	u	l	6	D
	v	r	g	z	n	9

Exemple :

Pour **msg="promotion bac 2019"** et le mot-clé "**CHAISE**"

Etape1 : La matrice **M1** est remplie aléatoirement comme suit :

Etape2 : Le résultat du chiffrement du message msg, caractère par caractère donne :

Caractère à chiffrer	p	r	o	m	o	t	i	o	n		b	a	c		2	0	1	9
Résultat du chiffrement	DA	FB	AC	BB	AC	BC	CA	AC	FE		BE	CC	AA		CD	DC	AB	FF

D'où le message **msgi** est le suivant : "DAFBACBBACBCCAACFE BECCAA CDDCABFF"

Etape3 : Remplissage de la matrice **M2**

Etape4 : Tri de la matrice **M2**

C	H	A	I	S	E
D	A	F	B	A	C
B	B	A	C	B	C
C	A	A	C	F	E
	B	E	C	C	A
A		C	D	D	C
A	B	F	F		

A	C	E	H	I	S
F	D	C	A	B	A
A	B	C	B	C	B
A	C	E	A	C	F
E		A	B	C	C
C	A	C		D	D
F	A		B	F	

Etape5 : Le message chiffré final est "FAAECFDBC AACCEAC ABAB BBCCCDFABFCD "

On se propose d'écrire un programme permettant :

- de saisir un message **msg** ne dépassant pas **18** caractères et formé uniquement de lettres minuscules, de chiffre et d'espace.
- de saisir un mot-clé formé de **6** lettres majuscules.
- de crypter le message **msg** selon la méthode de chiffrement décrite précédemment.

NB : On dispose d'un module **Initialisation (M1)** qui permet de remplir aléatoirement la matrice carrée **M1** comme décrit dans l'étape1 et que le candidat peut appeler dans sa solution sans le développer.

-D'afficher le message chiffré final.

Travail demandé :

1. Analyser le problème en le décomposant en modules.
2. Ecrire les algorithmes des modules envisagés.