Consignes 1. Les calculatrices sont interdites. Numérotez vos feuilles et faites apparaître les questions dans l'ordre de l'énoncé. Si vous repérez une erreur d'énoncé, signalez le sur votre copie et poursuivez votre composition.

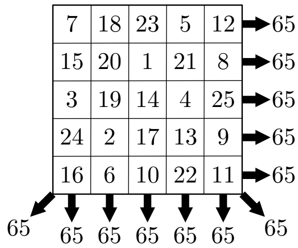

Figure 1

16	24	3	15	7
6	2	19	20	18
10	17	14	1	23
22	13	4	21	5
11	9	25	8	12

Figure 2

D				A
	D		A	
		A/D		
	A		D	
A				D

Figure 3

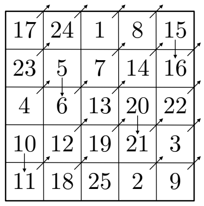

Figure 4

			5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 5

7	18	23	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 6

7	23	18	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 7

18	7	23	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 8

18	23	7	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 9

23	7	18	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 10

23	18	7	5	12
15	20	1	21	8
3	19	14	4	25
24	2	17	13	9
16	6	10	22	11

Figure 11

$c+a$	$c-a-b$	$c+b$
$c-a+b$	c	$c+a-b$
$c-b$	$c+a+b$	$c-a$

Figure 12

1	2
3	4

Figure 13

-2	-1
0	1

Figure 14

3	6
9	12

Figure 15

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

Figure 16

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

Figure 17

2	9	4
7	5	3
6	1	8

Figure 18

2	7	6
9	5	1
4	3	8

Figure 19

1	6	5
8	4	0
3	2	7

Figure 20

9	54	45
72	36	0
27	18	63

Figure 21

9	9	9	54	54	54	45	45	45
9	9	9	54	54	54	45	45	45
9	9	9	54	54	54	45	45	45
72	72	72	36	36	36	0	0	0
72	72	72	36	36	36	0	0	0
72	72	72	36	36	36	0	0	0
27	27	27	18	18	18	63	63	63
27	27	27	18	18	18	63	63	63
27	27	27	18	18	18	63	63	63

Figure 22

11	18	13	56	63	58	47	54	49
16	14	12	61	59	57	52	50	48
15	10	17	60	55	62	51	46	53
74	81	76	38	45	40	2	9	4
79	77	75	43	41	39	7	5	3
78	73	80	42	37	44	6	1	8
29	36	31	20	27	22	65	72	67
34	32	30	25	23	21	70	68	66
33	28	35	24	19	26	69	64	71

Figure 23

Définitions. Soit $n \in \mathbb{N}^*$. On appelle ***grille d'ordre*** n un tableau composé de n lignes et n colonnes. Chaque case de la grille est repérée par un couple (i, j) où i est l'indice de la ligne ($i = 0$ pour la ligne du haut, $i = n-1$ pour la ligne du bas) et j est l'indice de la colonne ($j = 0$ pour la colonne de gauche, $j = n-1$ pour la colonne de droite). Par exemple, la grille de la figure 1 est d'ordre $n = 5$, sa case de coordonnées $(1, 3)$ contient 21 et sa case de coordonnées $(4, 2)$ contient 10.

En Python, le type « `list[list[int]]` » sera noté « `grille` » et servira à représenter des grilles contenant des entiers. Ainsi, si « `G: grille` » correspond à une grille d'ordre n, alors `G` contient n sous-listes dont chacune est de taille n. Par exemple, la grille de la figure 1 correspond à :

```
G_fig1 = [[ 7, 18, 23,  5, 12], [15, 20,  1, 21,  8], [ 3, 19, 14,  4, 25],
          [24,  2, 17, 13,  9], [16,  6, 10, 22, 11]]
```

Un ***carré magique normal*** est une grille d'ordre n, où chaque entier de $[\![1; n^2]\!]$ apparaît une et seule fois, et telle que les sommes des entiers sur chaque ligne, chaque colonne et chacune des deux diagonales soient égales à $\dfrac{n(n^2+1)}{2}$. Par exemple, la figure 1 est un carré magique normal d'ordre $n = 5$.

Consignes 2.

- ⋆ Jusqu'à la fin du sujet, et même si ce n'est pas précisé, l'entier n est strictement positif et correspond au nombre de sous-listes dans la variable « `G: grille` ».
- ⋆ Lorsque vous écrivez une fonction qui prend en entrée une liste `L`, cette fonction ne doit pas modifier `L`. Bien sûr, vous avez le droit de modifier les listes ayant été créées à l'intérieur de la fonction.
- ⋆ Lorsque vous créez deux listes, vous devez faire en sorte qu'elles soient modifiables indépendamment. En d'autres termes, si on note `L` et `M` ces listes, alors `id(L)` et `id(M)` doivent être différents. En particulier, cette consigne est valable lorsque `L` et `M` sont deux sous-listes d'une même liste de listes.

I. Les cas $n = 1$, $n = 2$ et $n = 3$

1. (a) Déterminer en le justifiant le nombre de carrés magiques normaux d'ordre $n = 1$.
   (b) Même question pour $n = 2$.

Grilles de Lucas. Soient $(a, b, c) \in \mathbb{Z}^3$ trois entiers relatifs vérifiant la condition :

$$(C) : 0 < a < b < c - a \text{ et } b \neq 2a$$

La grille de Lucas associée à (a, b, c) est la grille d'ordre 3 présentée figure 12.

2. (a) Déterminer un triplet (a, b, c) pour lequel la grille de Lucas est un carré magique normal. On donnera le triplet, ainsi que la grille correspondante.
   (b) Écrire une fonction `Lucas(a: int, b: int, c: int) -> grille` qui renvoie la grille de Lucas associée à (a, b, c). Si la condition (C) n'est pas respectée, votre fonction déclenchera une erreur.

II. Test de magicité

L'objectif de cette partie est de tester si une variable « `G: grille` » représente un carré magique normal.

II.1. Contraintes sur les tailles et les valeurs de la grille

Rappel : l'entier n désigne le nombre de sous-listes dans `G`.

3. (a) Écrire une fonction `testTaille` qui prend en entrée `G`, renvoie `True` si toutes ses sous-listes sont de taille n et `False` sinon.
   (b) Donner la signature de la fonction `testTaille`.
4. Écrire une fonction `testVal1` qui prend en entrée `G`, renvoie `True` si tous les éléments de `G` appartiennent à $[\![1, n^2]\!]$ et `False` sinon. Attention, on ne suppose pas que `testTaille(G)` s'évalue en `True`.

Afin de vérifier que chaque entier de $[\![1, n^2]\!]$ appartient une et une seule fois à `G`, on propose la procédure suivante :

→ Pour chaque $k \in [\![1, n^2]\!]$:
  - On parcourt tous les éléments de `G` afin de calculer c_k le nombre d'éléments de `G` égaux à k.
  - Si $c_k \neq 1$, on arrête la fonction et on renvoie `False`.

→ Si tous les tours de boucle se sont exécutés sans renvoyer `False`, on renvoie `True`.

5. Supposons que `testTaille(G)` s'évalue en `True`. Donner en la justifiant la complexité de la procédure ci-dessus en fonction de n. Votre réponse devra être de la forme "Le temps d'exécution de cette procédure est en $\mathcal{O}(f(n))$" avec f une fonction bien choisie.

La complexité trouvée à la question 5 n'étant pas satisfaisante, on propose une autre procédure pour tester si chaque élément de $[\![1, n^2]\!]$ appartient à `G`. Pour cela, on construit une liste « `T: list[bool]` » de taille n^2 telle que pour chaque $i \in [\![1; n^2]\!]$, le booléen `T[i-1]` vaut `True` si `i` apparaît dans `G` et `False` sinon.

6. Attention, dans cette question `testTaille(G)` et `testVal1(G)` ne sont pas nécessairement égaux à `True`.
   (a) Écrire une fonction `getT(G: grille) -> list[bool]` qui renvoie la liste `T` décrite ci-dessus.
   (b) En déduire une fonction `testVal2(G: grille) -> bool` qui renvoie `True` si tous les élément de $[\![1, n^2]\!]$ apparaissent au moins une fois dans `G` et `False` sinon. La fonction `testVal2` devra avoir la même complexité que la fonction `getT`.
7. Supposons que `testTaille(G)` s'évalue en `True`. Donner la complexité de `testVal2` en fonction de n avec une justification détaillée.

À partir de maintenant, on pourra supposer sans le vérifier que `testTaille(G)`, `testVal1(G)` et `testVal2(G)` s'évaluent en `True`.

II.2. Somme des lignes, des colonnes et des diagonales

Dans cette partie, n représente toujours le nombre de sous-listes dans `G` et on pose :

$$N = \frac{n(n^2+1)}{2}.$$

Une grille peut satisfaire trois propriétés :

(L) : Pour chaque ligne de G, la somme des éléments est égale à N.
(C) : Pour chaque colonne de G, la somme des éléments est égale à N.
(D) : Pour chacune des deux diagonales de G, la somme des éléments est égale à N.

Par définition, une grille est un carré magique si elle satisfait (L), (C) et (D).

8. (a) Écrire une fonction `testL(G: grille) -> bool` qui renvoie `True` si `G` satisfait (L) et `False` sinon.
   (b) Expliquer comment modifier la fonction précédente pour tester la condition (C).

Pour tester la condition (C), on va utiliser une autre méthode que celle donnée à la question 8b. Notre but est de réutiliser la fonction `testL` déjà écrite. Étant donnée une grille « `G1: grille` », on note « `G2: grille` » la grille obtenue après avoir appliqué une rotation d'un quart de tour dans le sens de aiguilles d'une montre. Par exemple, si `G1` correspond à la figure 1, alors `G2` correspond à la figure 2. On remarque alors que `G1` vérifie (C) si et seulement si `G2` vérifie (L).

9. (a) Pour chacun des quatre entiers k se trouvant dans l'un des coins de `G1`, donner les coordonnées de k dans `G1` et dans `G2`.
   (b) Écrire une une fonction `rot(G1: grille) -> grille` qui prend en entrée `G1` et renvoie `G2`.
   (c) En déduire une fonction `testC(G: grille) -> bool` qui renvoie `True` si `G` satisfait (C) et `False` sinon. Vous devez utiliser les fonctions `rot` et `testL`.

Sur une grille, il y a deux diagonales, qu'on appellera ***diagonale*** et ***anti-diagonale***. Dans la grille de la figure 3, les cases se trouvant sur la diagonale (resp. anti-diagonale) sont marquées par un D (resp. A). En particulier, la case centrale est à la fois sur la diagonale et l'anti-diagonale.

10. Intéressons nous d'abord à la diagonale.
    (a) Considérons une case de coordonnées (i, j). Donner sans justification une condition nécessaire et suffisante sur i et j pour que la case soit sur la diagonale.
    (b) Écrire une fonction `testD1(G: grille) -> bool` qui renvoie `True` si la somme des éléments sur la diagonale de la grille est égale à N. Dans le cas contraire, votre fonction renverra `False`. Cette fonction devra être de complexité linéaire en n.
11. Passons maintenant aux anti-diagonales de la grille.
    (a) Refaire la question 10a dans le cas des anti-diagonales.
    (b) À l'aide d'une fonction **récursive**, refaire la question 10b dans le cas des anti-diagonales. La fonction obtenue devra être de complexité $\mathcal{O}(n)$ et de signature `testD2(G: grille) -> bool`.
12. Écrire une fonction `estCMN(G: grille) -> bool` qui renvoie `True` si `G` est un carré magique normal et `False` sinon.

III. Construction de carrés magiques

III.1. Méthode siamoise

Soit $n \in \mathbb{N}^*$ un entier impair. La méthode siamoise permet de construire un carré magique normal d'ordre n. Pour cela, on part d'une grille initialement vide, puis on y place les nombres de 1 à n^2 de la manière suivante :

→ L'entier 1 est placé dans la case se trouvant au milieu de la première ligne.

→ Pour chaque $k \in [\![1, n^2 - 1]\!]$, après avoir placé k dans une case c, on s'intéresse à la case se trouvant au nord-est de c. Si cette case est vide, on y place l'entier $k + 1$, sinon on place $k + 1$ dans la case se trouvant au sud de c.

→ Lorsqu'on se déplace sur la grille, on considère que celle-ci est circulaire : les bords haut/bas sont reliés ainsi que les bords droit/gauche.

Par exemple, pour $n = 5$ on obtient la figure 4.

13. Écrire une fonction `siamoise` qui prend en entrée un entier $n \in \mathbb{N}^*$ et renvoie le carré magique obtenu avec la méthode siamoise. Dans le cas où n est pair, votre fonction déclenchera une erreur.

III.2. Produit de carrés magiques

Soit G une grille d'ordre $n \in \mathbb{N}^*$ (G n'est pas nécessairement un carré magique) et $m \in \mathbb{N}^*$ un entier.

- ⋆ On note $G - m$ la grille d'ordre n dans laquelle chaque case c a été remplacée par une case contenant $k - m$ où k est le contenu de c. Par exemple, si G est la figure 13 alors $G - 3$ est la figure 14 ; et si G est la figure 19 alors $G - 1$ est la figure 20.
- ⋆ On note $m \times G$ la grille d'ordre n dans laquelle chaque case c a été remplacée par une case contenant $m \times k$ où k est le contenu de c. Par exemple, si G est la figure 13 alors $3 \times G$ est la figure 15 ; et si G est la figure 20 alors $9 \times G$ est la figure 21.
- ⋆ On note $f_1(G, m)$ la grille d'ordre nm consistant à juxtaposer m^2 copies de G. Par exemple, si G est la figure 13 alors $f_1(G, 3)$ est la figure 16.
- ⋆ On note $f_2(G, m)$ la grille d'ordre nm consistant à remplacer chaque case c de G par m^2 cases contenant k où k est la valeur de c. Par exemple, si G est la figure 13 alors $f_2(G, 3)$ est la figure 17 ; et si G est la figure 21 alors $f_2(G, 3)$ est la figure 22.

14. (a) Écrire une fonction `f1(G: grille, m: int) -> grille` qui renvoie $f_1(G, m)$.

    (b) Écrire une fonction `f2(G: grille, m: int) -> grille` qui renvoie $f_2(G, m)$.

Soit G_1 une grille d'ordre n_1 et G_2 une grille d'ordre n_2. Le produit de G_1 et G_2 est une grille G d'ordre $n_1 n_2$. Pour la construire, on définit :

$$H_1 = f_1(G_1, n_2) \qquad\qquad H_2 = f_2\Big(n_1^2 \times (G_2 - 1), n_1\Big)$$

La grille G est obtenue en sommant H_1 et H_2 case par case. Par exemple, si G_1 est la figure 18 et G_2 est la figure 19, alors $G_2 - 1$ est la figure 20, $n_1^2 \times (G_2 - 1)$ est la figure 21, H_2 est la figure 22 et G est la figure 23.

15. Écrire une fonction `produit(G1: grille, G2: grille) -> grille` qui renvoie le produit de deux grilles.

Remarque. On peut montrer que le produit de deux carrés magiques est un carré magique.

IV. Complétion de grilles

Soit G une grille partiellement remplie. Notre but est de générer tous les carrés magiques normaux pouvant être obtenus en complétant G. Par exemple, dans la figure 5, il manque les trois entiers 1, 18 et 23. Il y a alors 6 manières de compléter cette grille (voir les figures 6 à 11), mais seule la complétion de la figure 6 donne un carré magique normal.

16. Écrire une fonction `genCMN(G0: grille) -> list[grille]` qui prend en entrée la grille `G0` et renvoie la liste des carrés magiques normaux qu'il est possible d'obtenir en complétant `G0`. Dans cette question, on prend la convention que les cases vide de la grille correspondent à un 0 dans `G0`.