OLYMPIADES MATHÉMATIQUES
LYCÉE, PREMIÈRE

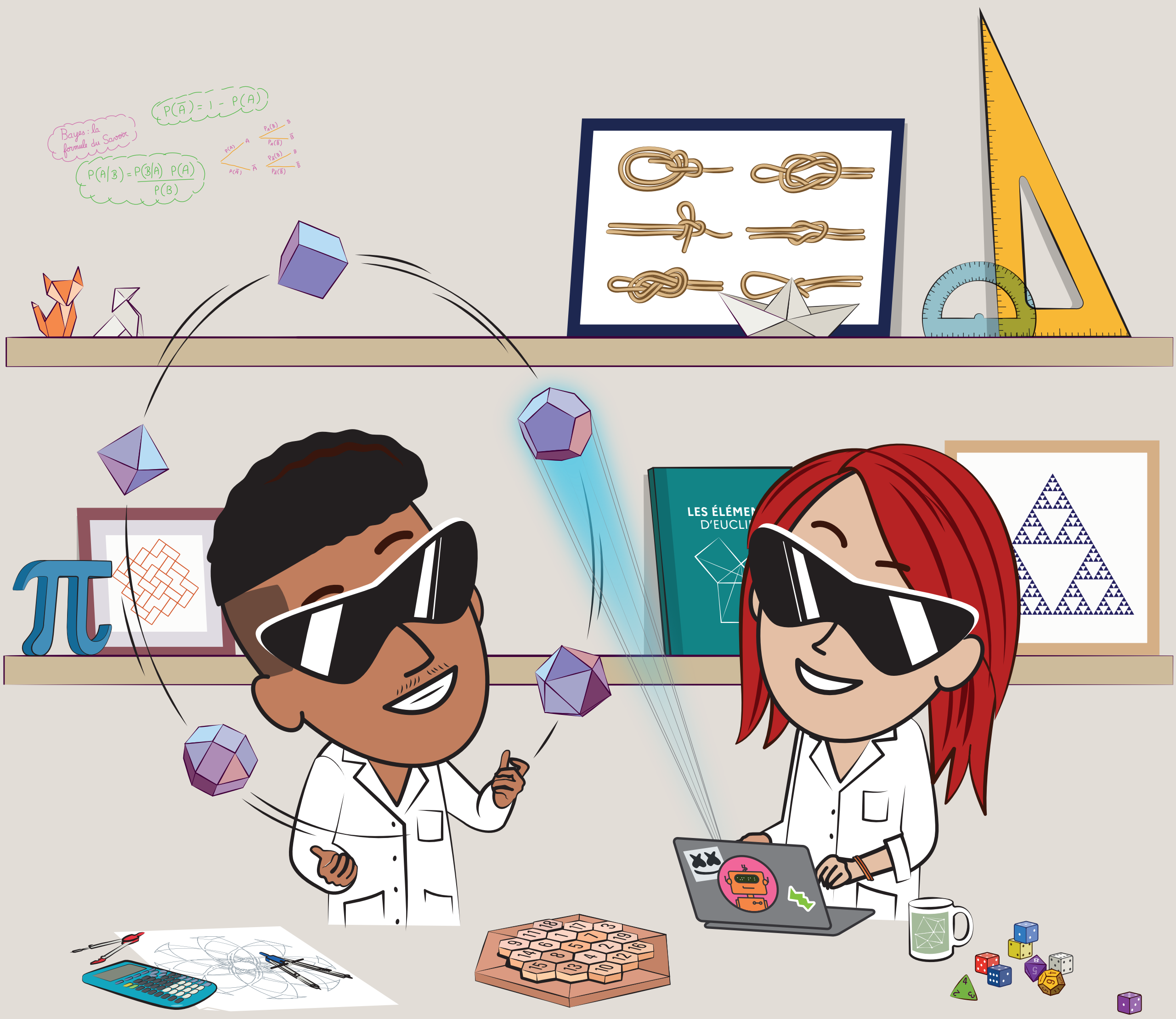
ACADÉMIE DE TOULOUSE
2023

CORRIGÉ DE L'ÉPREUVE

MINISTÈRE
DE L'ÉDUCATION
NATIONALE
ET DE LA JEUNESSE

Liberté
Égalité
Fraternité

OLYMPIADES NATIONALES 2023
DE MATHÉMATIQUES



Épreuve le mercredi 15 mars 2023 (le 14 en Polynésie française). 4 exercices en 4 heures.
Palmarès national et académiques, en individuel et par équipes mixtes (de 2, 3 ou 4), selon cursus (technologique, général, général spécialité mathématiques).
Inscriptions auprès de vos professeurs de mathématiques jusqu'au 17 février 2023.

CORRECTION OLYMPIADES

EXERCICES ACADÉMIQUES

TOULOUSE 2023

Exercice 1 : Le tournoi de l'Olympe

Partie 1 : Grand gagnant / Tournoi indécis

1.a. Le joueur A3 gagne tous ses matchs contre tous ses adversaires.

Il est donc « grand gagnant ».

A1	A2
A1	A3
A2	A3

Le tournoi n'est pas indécis.

1.b. Lorsque $n = 3$, le tournoi se compose de trois matchs. Chaque match peut avoir deux issues, suivant le gagnant du match. Le nombre de bilans possibles est donc égal à $2 \times 2 \times 2 = 8$.

Nous donnons ci-dessous les huit bilans possibles (en vert le gagnant du match, en rouge le perdant) ; parmi eux, deux bilans sont indécis.

A1	A2
A1	A3
A2	A3
A1 GG	

A1	A2
A1	A3
A2	A3
A1 GG	

A1	A2
A1	A3
A2	A3
Indécis	

A1	A2
A1	A3
A2	A3
A3 GG	

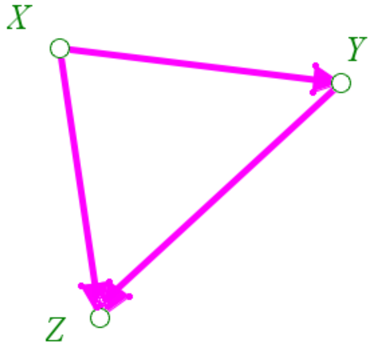
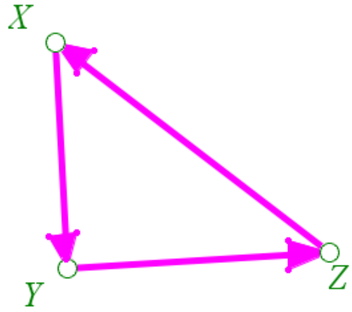
A1	A2
A1	A3
A2	A3
A2 GG	

A1	A2
A1	A3
A2	A3
Indécis	

A1	A2
A1	A3
A2	A3
A2 GG	

A1	A2
A1	A3
A2	A3
A3 GG	

En conclusion, nous aboutissons à l’un ou l’autre des deux types de configuration :

 L’un des joueurs est « grand gagnant ». Un autre joueur perd ses deux matchs.	 Le tournoi est indécis.

2. Soit *n* un entier au moins égal à 3.

Un match est déterminé par une paire de joueurs. Le nombre de matchs à organiser est égal au nombre de paires, donc de parties à 2 éléments, d’un ensemble à *n* éléments.

Ce nombre de matchs est donc égal au nombre de combinaisons de 2 éléments dans un ensemble à *n* éléments.

Il s’agit du nombre :

$$\binom{n}{2} = \frac{n \times (n-1)}{2}$$

Un match donné pouvant avoir deux issues possibles, suivant le joueur gagnant du match, le nombre de bilans possibles est le nombre :

$$2^{\frac{n \times (n-1)}{2}}$$

3.a. Vrai. Si un des joueurs (A_1 par exemple) gagne tous ses matchs contre tous ses adversaires, il est « grand gagnant » quels que soient les résultats des matchs des joueurs A_2 à A_n entre eux.

3.b. Vrai. Le bilan du tournoi où pour tout *i* tel que $1 \leq i \leq n-1$, le joueur A_i gagne contre A_{i+1} et où A_n gagne contre A_1 (chaque joueur gagne contre son suivant et le dernier joueur gagne contre le premier) est indécis car chaque joueur subit au moins une défaite, il n’y a pas de « grand gagnant ».

Partie 2 : Petits gagnants

4.a. Justifions que A_2 est « petit gagnant » :

D'après le graphe de l'énoncé,

- A_2 gagne contre A_3 et A_5.
- A_2 perd contre A_1 mais gagne contre A_3 qui gagne contre A_1.
- A_2 perd contre A_4 mais gagne contre A_3 qui gagne contre A_4.

Pour chaque adversaire de A_2 une des deux conditions pour être « petit gagnant » est satisfaite.

Nous avons vérifié que A_2 est « petit gagnant ».

4.b. Cherchons les autres « petits gagnants » :

A_1 n'est pas « petit gagnant » car il perd contre A_4 et le seul joueur battu par A_1 (le joueur A_2) perd contre A_4.

A_5 n'est pas « petit gagnant » non plus car il perd contre A_4 et le seul joueur battu par A_5 (le joueur A_1) perd contre le joueur A_4.

A_3 est « petit gagnant ». En effet :

- A_3 gagne contre A_1, A_4 et A_5.
- A_3 perd contre A_2 mais gagne contre A_1 qui gagne contre A_2.

A_4 est aussi « petit gagnant ». En effet :

- A_4 gagne contre A_1, A_2 et A_5.
- A_4 perd contre A_3 mais gagne contre A_2 qui gagne contre A_3.

Les « petits gagnants » du tournoi sont A_2, A_3 et A_4.

5.a. Dans la question précédente, nous avons trouvé trois « petits gagnants ». Les joueurs A_3 et A_4 comptent trois victoires, tandis que A_2 ne compte que deux victoires.

Ainsi, A_2 est un « petit gagnant » tout en ne comptant pas le plus grand nombre de victoires.

Dans le cas où $n = 5$, il est faux de dire qu'un « petit gagnant » a le plus grand nombre de victoires.

5.b. Considérons le tournoi étudié à la **question 4** avec cinq joueurs, où l'on a un « petit gagnant » qui n'a pas le plus grand nombre de victoires. Ajoutons un sixième joueur et supposons que ce nouveau joueur perde tous ses matches face à ses 5 adversaires. Les « petits gagnants » restent les mêmes, avec chacun une victoire de plus remportée face à ce nouveau joueur (qui n'est pas un « petit gagnant » puisqu'il perd tous ses matchs). Il y a au moins parmi les 6 joueurs de ce tournoi un petit gagnant qui n'a pas le plus grand nombre de victoires.

Dans le cas où $n = 6$, il est faux de dire qu'un « petit gagnant » a le plus grand nombre de victoires.

En réitérant ce raisonnement « de proche en proche », consistant à ajouter un par un des joueurs perdant leurs matchs, on démontre que pour tout entier $n \geq 6$, il est faux de dire qu'un « petit gagnant » a le plus grand nombre de victoires.

NB. Il nous semble qu'au niveau d'une classe de Première, l'argument « de proche en proche » est suffisant. Ultérieurement, il sera utile de mettre en place un « raisonnement par récurrence » raisonnement[1] dont le principe n'est enseigné qu'en Terminale. Un tel raisonnement n'est donc pas exigible au niveau Première.
Nous en proposons une mouture ci-dessous :

Considérons la propriété $\wp_n$:
« Dans le cas de *n* joueurs, il est faux de dire qu'un « petit gagnant » a le plus grand nombre de victoires ».

- **Initialisation :** La question **5.a** a prouvé que $\wp_5$ est vraie.

- **Hérédité :** Supposons que, pour un certain entier *n*, $\wp_n$ soit vraie. Il existe un bilan de tournoi entre *n* joueurs où l'un au moins des joueurs est « petit gagnant » mais n'a pas le plus grand nombre de victoires. Ajoutons un nouveau joueur A_{n+1} qui perd tous ses matchs face à tous ses adversaires A_1, …, A_n. Les « petits gagnants restent les mêmes, avec chacun une victoire de plus remportée face à ce nouveau joueur (qui n'est pas un « petit gagnant » puisqu'il perd tous ses matchs). Il y a au moins parmi les $n+1$ joueurs un petit gagnant qui n'a pas le plus grand nombre de victoires.
Pour cet entier $n+1$, il est donc faux de dire qu'un « petit gagnant » a le plus grand nombre de victoires.
Ainsi : $\wp_n \implies \wp_{n+1}$.
- **Conclusion :**

La propriété $\wp_n$ étant héréditaire et initialisée au rang 5, elle est vraie pour tout entier $n \geq 5$.

6. Raisonnons « par l'absurde » comme l'énoncé le suggère.
Supposons que *A* soit un joueur qui a le plus grand nombre de victoires mais qui n'est pas un « petit gagnant ».

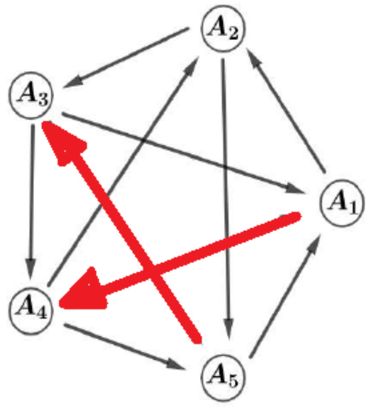
Cela signifie qu'il existerait un joueur *B* qui a gagné contre *A* et tel qu'aucun des joueurs battus par *A* n'a gagné contre *B*. En conséquence, *B* aurait gagné contre *A* et aussi contre tous les joueurs battus par *A* : ce joueur *B* aurait au moins une victoire de plus que *A*.
Ce qui contredirait l'hypothèse : « le joueur *A* a le plus grand nombre de victoires ».

Nécessairement, un joueur *A* qui a le plus grand nombre de victoires est aussi un « petit gagnant ».

[1] Voir aussi sur ce thème l'article Wikipedia

7.a. La réponse est oui. Pour $n = 3$, il y a des bilans de tournois où tous les joueurs sont des « petits gagnants » : dans le cas des deux tournois indécis que nous avons étudiés dans la **question 1.b**, chaque joueur est un « petit gagnant ».

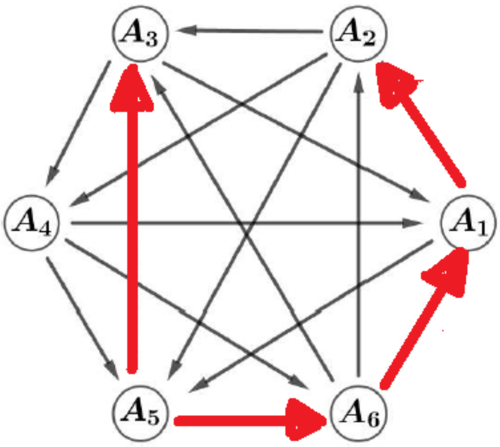
7.b. Nous avons repris le tournoi proposé par l'énoncé en **question 4** mais en inversant le résultat de deux matches (flèches rouges). Les joueurs A_1 et A_5 deviennent « petits gagnants » sans qu'aucun autre joueur ne perde ce statut.



7.c. Nous proposons le tournoi ci-contre.

Le joueur A_5 doit nécessairement gagner contre A_3 et contre A_6 puisqu'il perd contre A_4.

Le joueur A_1 gagne contre A_2 puisqu'il perd contre A_4 et de ce fait, A_6 doit gagner contre A_1.



7.d. Etant donné un tournoi de n joueurs où tous les joueurs de A_1 à A_n sont « petits gagnants », ajoutons deux joueurs A_{n+1} et A_{n+2}.

Pour tout entier i tel que $1 \leq i \leq n$, construisons un tournoi « indécis » entre les trois joueurs A_i, A_{n+1} et A_{n+2} : En supposant par exemple que A_{n+1} gagne contre A_{n+2}, faisons en sorte que A_i gagne contre A_{n+1} et que A_{n+2} gagne contre A_i. Les joueurs A_1 à A_n restent « petits gagnants », tandis que A_{n+1} et A_{n+2} sont eux aussi petits gagnants en vertu de la deuxième condition permettant d'être « petit gagnant ».

S'il existe un tournoi de n joueurs où tous les joueurs sont « petits gagnants », alors il existe aussi un tournoi de $n + 2$ joueurs où tous les joueurs sont « petits gagnants ».

En itérant le procédé consistant à ajouter des joueurs de deux en deux, il en résulte que s'il existe un tournoi de n joueurs où tous les joueurs de A_1 à A_n sont « petits gagnants », alors pour tout entier strictement positif k, il existe un tournoi de $n + 2k$ joueurs où tous les joueurs de A_1 à A_{n+2k} sont « petits gagnants ».

8. Puisqu'il existe un bilan de tournoi à 5 joueurs où tous les joueurs de A_1 à A_5 sont « petits gagnants », alors pour tout entier strictement positif k, il existe un tournoi de $5+2k$ joueurs où tous les joueurs de A_1 à A_{5+2k} sont « petits gagnants » (on englobe ainsi tous les indices impairs supérieurs ou égaux à 5).
Puisqu'il existe un bilan de tournoi à 6 joueurs où tous les joueurs de A_1 à A_6 sont « petits gagnants », alors pour tout entier strictement positif k, il existe un tournoi de $6+2k$ joueurs où tous les joueurs de A_1 à A_{6+2k} sont « petits gagnants » (on englobe ainsi tous les indices pairs supérieurs ou égaux à 6).

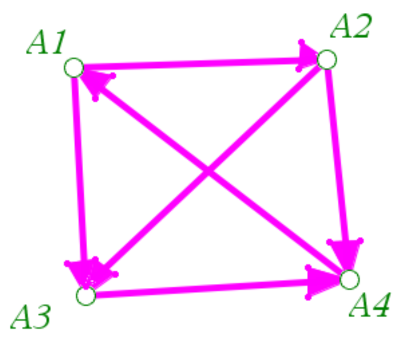
En définitive, il existe un bilan de tournoi à n joueurs où tous les joueurs de A_1 à A_n sont « petits gagnants » pour tout entier $n \geq 5$, quelle que soit sa parité. Nous avons vu que, pour $n = 3$, c'était aussi le cas.

9. Cherchons à construire un tournoi à 4 joueurs où tous les joueurs sont petits gagnants.
Dans le cas de 4 joueurs, chacun d'eux joue 3 matchs, contre chacun des 3 autres joueurs.

- Si un des joueurs gagne tous ses matchs, il est « grand gagnant » et aucun des autres joueurs ne peut être un « petit gagnant ».
- Si un des joueurs perd tous ses matchs, il ne peut être un « petit gagnant ».

Il reste à examiner le cas où deux joueurs (par exemple A_1 et A_2) gagnent deux matchs et les deux autres (A_3 et A_4 en l'occurrence) en gagnent un seul.
Supposons que A_1 gagne contre A_2. Alors A_2 gagne contre A_3 et contre A_4 (puisqu'il gagne deux matchs) tandis que A_1 gagne contre l'un de ces deux joueurs (par exemple A_3) et perd contre l'autre, en l'occurrence A_4. Enfin, A_3 gagne contre A_4 (puisque A_3 et A_4 gagnent chacun un match).

Nous aboutissons au tournoi ci-contre. Dans cette configuration, A_3 n'est pas un « petit gagnant » puisqu'il perd contre A_2 et que le seul joueur contre lequel il gagne, le joueur A_4, perd contre A_2.



Il n'y a donc aucun tournoi dans lequel les 4 joueurs sont tous des petits gagnants.

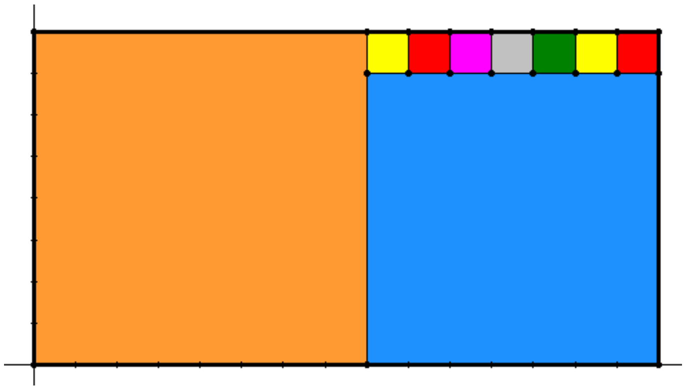
Exercice 2 : Le dernier carré

Partie A

1. Suites associées à un rectangle

1.a. Le rectangle de format 15×8 est découpé en un carré de côté 8, un carré de côté 7 et sept carrés de côté 1.

La suite qui lui est associée est la suite $1 - 1 - 7$.



1.b … avec Python

Nous avons écrit un algorithme Python nommé **découpage** qui a pour arguments les dimensions *a* et *b* du rectangle initial (supposées être des entiers strictement positifs). Cet algorithme décrit les différentes étapes du découpage à réaliser, jusqu'à ce que le rectangle soit complètement découpé en carrés (on conjecture que ce découpage complet est possible, ce que nous démontrons un peu plus loin).

```
>>> def decoupage(a,b):
        long=max(a,b)
        larg=min(a,b)
        while larg>0:
                u=long
                v=larg
                long=max(u-v,v)
                larg=min(u-v,v)
                if larg>0:
                        print("Découper un carré de côté",v,"; il reste un morceau",long,"x",larg)
                else:
                        print("Découper un carré de côté",v,"et c'est fini")
```

Test de l'algorithme sur l'exemple de l'énoncé.	>>> decoupage(11,4) Découper un carré de côté 4 ; il reste un morceau 7 x 4 Découper un carré de côté 4 ; il reste un morceau 4 x 3 Découper un carré de côté 3 ; il reste un morceau 3 x 1 Découper un carré de côté 1 ; il reste un morceau 2 x 1 Découper un carré de côté 1 ; il reste un morceau 1 x 1 Découper un carré de côté 1 et c'est fini
Application au cas d'un rectangle de format 15×8. Nous retrouvons le fait que la suite qui lui est associée est la suite $1 - 1 - 7$.	>>> decoupage(15,8) Découper un carré de côté 8 ; il reste un morceau 8 x 7 Découper un carré de côté 7 ; il reste un morceau 7 x 1 Découper un carré de côté 1 ; il reste un morceau 6 x 1 Découper un carré de côté 1 ; il reste un morceau 5 x 1 Découper un carré de côté 1 ; il reste un morceau 4 x 1 Découper un carré de côté 1 ; il reste un morceau 3 x 1 Découper un carré de côté 1 ; il reste un morceau 2 x 1 Découper un carré de côté 1 ; il reste un morceau 1 x 1 Découper un carré de côté 1 et c'est fini

Cas d'un rectangle de format 158×49.
Ce rectangle est découpé en 3 carrés de côté 49, 4 carrés de côté 11, 2 carrés de côté 5 et 5 carrés de côté 1.
La suite qui lui est associée est la suite $3 — 4 — 2 — 5$.
	`>>> decoupage(158,49)`
	`Découper un carré de côté 49 ; il reste un morceau 109 x 49`
	`Découper un carré de côté 49 ; il reste un morceau 60 x 49`
	`Découper un carré de côté 49 ; il reste un morceau 49 x 11`
	`Découper un carré de côté 11 ; il reste un morceau 38 x 11`
	`Découper un carré de côté 11 ; il reste un morceau 27 x 11`
	`Découper un carré de côté 11 ; il reste un morceau 16 x 11`
	`Découper un carré de côté 11 ; il reste un morceau 11 x 5`
	`Découper un carré de côté 5 ; il reste un morceau 6 x 5`
	`Découper un carré de côté 5 ; il reste un morceau 5 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 4 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 3 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 2 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 1 x 1`
	`Découper un carré de côté 1 et c'est fini`

2. Rectangles associés à une suite

On suppose dans cette question que les plus petits carrés découpés sont des carrés de côté 1. On trouverait d'autres solutions en multipliant les dimensions du rectangle initial par un même entier strictement positif k.

2.a. Recollons les morceaux en considérant de droite à gauche les termes de la suite :

- On recolle 2 carrés de côté 1 pour obtenir un rectangle de format 2×1
- On recolle 1 carré de côté 2 pour obtenir un rectangle de format 3×2
- On recolle 3 carrés de côté 3 pour obtenir un rectangle de format 11×3

Un rectangle initial associé à la suite $3 — 1 — 2$ est le rectangle de dimensions 11×3.

2.b. Continuons à recoller les morceaux :

Nous avons obtenu un rectangle de format 11×3 à partir de la suite $3 — 1 — 2$. Continuons …

- On recolle 1 carré de côté 11 pour obtenir un rectangle de format 14×11
- On recolle 2 carrés de côté 14 pour obtenir un rectangle de format 39×14

Un rectangle initial associé à la suite $2 — 1 — 3 — 1 — 2$ est le rectangle de dimensions 39×14.

Vérifions ci-contre (facultativement) nos résultats à l'aide de notre algorithme Python.
L'algorithme « découpage » confirme bien nos résultats.
	`>>> decoupage(11,3)`
	`Découper un carré de côté 3 ; il reste un morceau 8 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 5 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 3 x 2`
	`Découper un carré de côté 2 ; il reste un morceau 2 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 1 x 1`
	`Découper un carré de côté 1 et c'est fini`
	`>>> decoupage(39,14)`
	`Découper un carré de côté 14 ; il reste un morceau 25 x 14`
	`Découper un carré de côté 14 ; il reste un morceau 14 x 11`
	`Découper un carré de côté 11 ; il reste un morceau 11 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 8 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 5 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 3 x 2`
	`Découper un carré de côté 2 ; il reste un morceau 2 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 1 x 1`
	`Découper un carré de côté 1 et c'est fini`

3. Suites impossibles

De façon générale, il n'est pas possible que le dernier terme d'une suite de plusieurs termes soit égal à 1. Cela signifierait que l'avant dernier découpage laisserait un unique carré. Mais dans ce cas, ce carré aurait pour côté le même que ceux de l'avant dernier découpage. Il ne s'agirait pas d'un nouveau type de carré. Toute suite de plusieurs termes se terminant par un « 1 », en particulier la suite $4 - 3 - 2 - 1$, est une suite impossible.

4. Cas où longueur *a* et largeur *b* sont des nombres entiers

NB. Nous proposons deux résolutions, une résolution strictement conforme au programme de Première en vigueur (sans l'algorithme d'Euclide) puis une résolution avec l'algorithme d'Euclide, outil opportun dans ce contexte mais qui ne figure qu'au programme Maths Expertes de Terminale.

Solution A. Sans l'algorithme d'Euclide (conforme au programme de Première)

Soit un rectangle de longueur L et de largeur l (on suppose $L > l$, car s'il y a égalité le rectangle est déjà un carré, il n'y a aucun découpage à faire).

Ecrivons la division euclidienne de L par l : $L = q \times l + r$, où q est le quotient et r est le reste de cette division ($0 \leq r < l$). Nous pouvons découper dans le rectangle initial q carrés de côté l. Si r est nul, le découpage est terminé. Sinon, il reste un rectangle de longueur l et de largeur r, des entiers naturels strictement plus petits que leurs homologues du rectangle initial ($l < L$ et $r < l$).

Itérons le procédé avec ce nouveau rectangle en effectuant la division euclidienne de sa longueur par sa largeur. Si le reste de cette division est nul, le découpage est terminé. Sinon, il restera un rectangle de dimensions entières strictement plus petites que leurs homologues du précédent rectangle.

De proche en proche, et tant que le reste de la division euclidienne effectuée est non nul, nous obtiendrons des rectangles résiduels dont la suite des longueurs et la suite des largeurs sont des suites d'entiers naturels strictement décroissantes. Or, une suite d'entiers naturels strictement décroissante est nécessairement finie : Au bout d'un nombre fini d'opérations (b opérations au plus), nous obtiendrons un reste nul et le procédé sera arrêté.

La suite associée au rectangle initial est nécessairement finie.

Solution B. Si vous êtes en Terminale option Maths Expertes vous pouvez résoudre cette question en utilisant l'algorithme d'Euclide comme il suit :

Faisons un lien entre la suite associée à un rectangle de dimensions entières L et l et l'algorithme d'Euclide appliqué à ces deux entiers. Ecrivons cet algorithme : $\begin{cases} L = lq_1 + r_1 \\ l = r_1 q_2 + r_2 \\ r_1 = r_2 q_3 + r_3 \\ \dots \\ r_{n-1} = r_n q_{n+1} \end{cases}$ (le reste r_n étant le dernier non nul). Les égalités écrites étant des divisions euclidiennes : $0 < r_n < r_{n-1} < \cdots < r_1 < l$

(On note au passage que, sauf si $L = l$, l'entier q_{n+1} est > 1, sinon on aurait $r_{n-1} = r_n$, ce qui est impossible ; nous retrouvons le résultat de la **question 3**).

Le rectangle peut être découpé en q_1 carrés de côté b, q_2 carrés de côté r_1, …, q_{n+1} carrés de côté r_n.

Ainsi, la suite associée au rectangle est la suite $q_1 - q_2 - \cdots - q_{n+1}$ des quotients des divisions euclidiennes figurant dans l'algorithme d'Euclide appliqué aux entiers L et l.

Nous savons que l'algorithme d'Euclide appliqué à deux entiers strictement positifs quelconques met en œuvre un nombre fini de divisions, nous en déduisons que, quels que soient les entiers $L \geq l > 0$, la suite des quotients $q_1 - q_2 - \cdots - q_{n+1}$, associée au rectangle de dimensions $L \times l$, est une suite finie.

Partie B

NB. Dans cette partie, nous reprenons à notre compte l'algorithme des divisions successives décrit dans la **question A.4** à cela près que ces divisions s'appliquent maintenant à des réels strictement positifs.

De façon générale, si *a* et *b* sont deux ***réels*** tels que $a \geq b > 0$, il existe un ***entier*** strictement positif *q* et un ***réel r*** **vérifiant** $\mathbf{0} \leq \boldsymbol{r} < \boldsymbol{b}$ tels que : $a = bq + r$.

- Divisons ainsi L par l : $L = lq_1 + r_1$
- Puis l par r_1 : $l = r_1 q_2 + r_2$
- Puis r_1 par r_2 : $r_1 = r_2 q_3 + r_3$
- Puis r_2 par r_3 …

Et ainsi de suite tant que l'on n'obtient pas un reste nul. Mais ces nombres r_i sont maintenant des *réels* et non plus des entiers. De ce fait, la suite (r_i) peut très bien ne pas être finie. L'algorithme ainsi construit peut « ne pas s'arrêter », il se peut qu'il n'existe aucun indice *n* tel que $r_{n+1} = 0$ et la suite $q_1 - q_2 - \cdots - q_n - \cdots$ associée à un rectangle peut ne pas être finie.

5. Formats particuliers associés à des suites finies

5.a. La suite associée à un rectangle $L \times l$ ne comporte qu'un terme si et seulement si l'algorithme d'Euclide appliqué à L et à *l* ne comporte qu'une ligne, c'est-à-dire si et seulement s'il existe un entier q_1 tel que : $L = lq_1$. Donc si et seulement si $\frac{L}{l} = q_1$ est un nombre entier.

Le format d'un rectangle dont la suite associée ne comporte qu'un terme est un nombre entier.

5.b. La suite associée à un rectangle $L \times l$ comporte deux termes si et seulement si l'algorithme d'Euclide appliqué à L et à l comporte deux lignes : $\begin{cases} L = lq_1 + r_1 \\ l = r_1 q_2 \end{cases}$. Soit : $\begin{cases} L = r_1 q_2 q_1 + r_1 \\ l = r_1 q_2 \end{cases}$.

Nous obtenons :

$$\frac{L}{l} = \frac{r_1 q_2 q_1 + r_1}{r_1 q_2} = q_1 + \frac{1}{q_2}$$

On peut écrire le format de ce rectangle sous la forme (q_1 et q_2 étant des entiers strictement positifs) :

$$f = q_1 + \frac{1}{q_2}$$

Exemple : le rectangle de dimensions $L = 13$; $l = 3$ qui a pour format : $$f = \frac{13}{3} = 4 + \frac{1}{3}$$	`>>> decoupage(13,3)`
	`Découper un carré de côté 3 ; il reste un morceau 10 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 7 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 4 x 3`
	`Découper un carré de côté 3 ; il reste un morceau 3 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 2 x 1`
	`Découper un carré de côté 1 ; il reste un morceau 1 x 1`
	`Découper un carré de côté 1 et c'est fini`

6. Format doré

6.a. Considérons un rectangle de largeur *l* et de longueur $L = f \times l$ où $1 < f < 2$.

Si on découpe un carré de côté *l*, le morceau restant est un rectangle de dimensions *l* et $L - l = (f - 1) \times l$.

L'hypothèse $1 < f < 2$ implique que $0 < f - 1 < 1$ et par conséquent que $L - l = (f - 1) \times l < l$.

Le morceau restant est donc un rectangle dont *l* est la longueur et dont $L - l = (f - 1) \times l$ est la largeur.

Le format *f'* de ce rectangle est le quotient de sa longueur par sa largeur. Nous obtenons :

$$f' = \frac{l}{L - l} = \frac{l}{(f - 1)l} = \frac{1}{f - 1}$$

6.b. Le format *f* est le format doré si et seulement si : $f = \frac{1}{f-1}$, c'est-à-dire si et seulement si $f \times (f - 1) = 1$ comme le demande l'énoncé, mais surtout, et plus efficacement, si et seulement si : $f^2 - f - 1 = 0$.

6.c. Le format doré est nécessairement une solution de l'équation au second degré $x^2 - x - 1 = 0$, équation qui a deux solutions : $\frac{1-\sqrt{5}}{2}$ (qui est strictement négative) et $\frac{1+\sqrt{5}}{2}$ (qui est strictement positive).

Ce format étant un réel strictement positif, il ne peut s'agir que de la solution strictement positive.

Le format doré est le nombre :

$$f_d = \frac{1+\sqrt{5}}{2}$$

6.d. Puisque, après découpage d'un carré, on obtient comme morceau restant un rectangle de même format que le rectangle initial, le processus de découpage se répète indéfiniment de la même manière.

La suite associée à un rectangle de format doré est la suite infinie constante, telle que $q_n = 1$ pour tout n.

Partie C

7. Formats associés à des suites infinies cycliques

7.a. Considérons le développement de l'algorithme lorsqu'il y a un cycle « $1 - 2$ » : $\begin{cases} L = l + r_1 \\ l = 2r_1 + r_2 \\ r_1 = r_2 + r_3 \\ r_2 = 2r_3 + r_4 \\ \dots \end{cases}$.

Désignons par f_1 le format du rectangle obtenu après le premier découpage d'un carré et par f_2 le format du rectangle obtenu après avoir découpé ensuite deux carrés identiques.

- Format du rectangle initial : $f = \frac{L}{l}$.
- Format du rectangle après le premier découpage : $f_1 = \frac{l}{r_1}$.
- Format du rectangle après le deuxième découpage : $f_2 = \frac{r_1}{r_2}$.

Déduisons de l'algorithme des relations entre f et f_2 :

- La relation $L = l + r_1$ implique que $\frac{L}{l} = 1 + \frac{r_1}{l}$ soit que $f = 1 + \frac{1}{f_1}$
- La relation $l = 2r_1 + r_2$ implique que $\frac{l}{r_1} = 2 + \frac{r_2}{r_1}$ soit que $f_1 = 2 + \frac{1}{f_2}$

Il en résulte la relation :

$$f = 1 + \frac{1}{f_1} = 1 + \frac{1}{2 + \frac{1}{f_2}} = \frac{1 + 3f_2}{1 + 2f_2}$$

Le cycle « $1-2$ » s'installe si et seulement si le format du rectangle obtenu après la deuxième étape de découpage est le même que le format du rectangle initial, soit si et seulement si $f = f_2$.

Le format f doit vérifier la relation : $f = \frac{1+3f}{1+2f}$ soit, de façon équivalente, la relation : $2f^2 - 2f - 1 = 0$

Ce format est nécessairement une solution de l'équation au second degré $2x^2 - 2x - 1 = 0$, équation qui a deux solutions : $\frac{1-\sqrt{3}}{2}$ (qui est strictement négative) et $\frac{1+\sqrt{3}}{2}$ (qui est strictement positive).
Ce format étant un réel strictement positif, il ne peut s'agir que de la solution strictement positive.
Ce format est le nombre :

$$f = \frac{1+\sqrt{3}}{2}$$

7.b. L'algorithme se présente ainsi : $\begin{cases} L = 5l + r_1 \\ l = 2r_1 + r_2 \\ r_1 = r_2 + r_3 \\ r_2 = 2r_3 + r_4 \\ \quad \dots \end{cases}$, le cycle « $1-2$ » s'installant à partir de troisième ligne.

- Format du rectangle initial : $f = \frac{L}{l}$.
- Format du rectangle après le premier découpage : $f_1 = \frac{l}{r_1}$.
- Format du rectangle après le deuxième découpage : $f_2 = \frac{r_1}{r_2}$

Déduisons de l'algorithme des relations entre f et f_2 :

- La relation $L = 5l + r_1$ implique que $\frac{L}{l} = 5 + \frac{r_1}{l}$ soit que $f = 5 + \frac{1}{f_1}$.
- La relation $l = 2r_1 + r_2$ implique que $\frac{l}{r_1} = 2 + \frac{r_2}{r_1}$ soit que $f_1 = 2 + \frac{1}{f_2}$.

Sachant que $f_2 = \frac{1+\sqrt{3}}{2}$, nous obtenons $f_1 = 2 + \frac{1}{\frac{1+\sqrt{3}}{2}} = 1 + \sqrt{3}$ et :

$$f = 5 + \frac{1}{1+\sqrt{3}} = \frac{9+\sqrt{3}}{2}$$

7.c.i. Conjecturons … avec Python :

Modifions l'algorithme **découpage**. Il a maintenant deux arguments, le format *f* et le nombre *n* de découpages que l'on veut faire. La boucle « while » est remplacée par une boucle « for ».	`>>> def decoupage(f,n):`
	`long=f`
	`larg=1`
	`for k in range(1,n+1):`
	`u=long`
	`v=larg`
	`long=max(u-v,v)`
	`larg=min(u-v,v)`
	`form=long/larg`
	`print("Découper un carré de côté",round(v,3),`
	`"; il reste un morceau de format",round(form,3))`

Appliquons cet algorithme au cas d'un rectangle de format $2\sqrt{2}$. Un nombre de découpages $n = 8$ nous paraît « a visto de nas » un nombre « raisonnable » pour être en mesure de conjecturer :

```
>>> from math import sqrt
>>> decoupage(2*sqrt(2),8)
Découper un carré de côté 1 ; il reste un morceau de format 1.828
Découper un carré de côté 1 ; il reste un morceau de format 1.207
Découper un carré de côté 0.828 ; il reste un morceau de format 4.828
Découper un carré de côté 0.172 ; il reste un morceau de format 3.828
Découper un carré de côté 0.172 ; il reste un morceau de format 2.828
Découper un carré de côté 0.172 ; il reste un morceau de format 1.828
Découper un carré de côté 0.172 ; il reste un morceau de format 1.207
Découper un carré de côté 0.142 ; il reste un morceau de format 4.828
```

Nous pouvons conjecturer que le format du rectangle obtenu après la sixième découpe est le même que celui du rectangle obtenu après la première découpe. Si c'est bien le cas, il s'installe un cycle « $1 - 4$ » à partir du rectangle obtenu après la première découpe.

7.c.ii. Démontrons :

Partant d'un rectangle $2\sqrt{2} \times 1$, le rectangle obtenu après la première découpe est un rectangle $\left(2\sqrt{2} - 1\right) \times 1$ de format $f = 2\sqrt{2} - 1$.

Après une deuxième découpe, nous obtenons un rectangle $1 \times \left(2\sqrt{2} - 2\right)$ dont 1 est maintenant la longueur.

Après une troisième découpe, nous obtenons un rectangle $\left(2\sqrt{2} - 2\right) \times \left(3 - 2\sqrt{2}\right)$ dont $2\sqrt{2} - 2$ est maintenant la longueur.

Découpons trois carrés de côté $3 - 2\sqrt{2}$.

Nous obtenons un rectangle de largeur $3 - 2\sqrt{2}$ dont la longueur est $\left(2\sqrt{2} - 2\right) - \left(3 - 2\sqrt{2}\right) = 8\sqrt{2} - 11$.

Calculons son format :

$$\frac{8\sqrt{2} - 11}{3 - 2\sqrt{2}} = \frac{\left(8\sqrt{2} - 11\right) \times \left(3 + 2\sqrt{2}\right)}{\left(3 - 2\sqrt{2}\right) \times \left(3 + 2\sqrt{2}\right)} = 2\sqrt{2} - 1$$

Nous avons bien retrouvé le même format que celui du rectangle obtenu après la première découpe. Nous pouvons conclure que la suite associée au rectangle initial est la suite $2 - 1 - 4 - 1 - 4 - \cdots$ avec un cycle « $1 - 4$ » qui se répète indéfiniment.

Exercice 3 : Nombres fléchés

Partie A

1.a. Dans la grille 1, le nombre 14 est décomposé deux fois de la même façon en $14 = 2 + 3 + 4 + 5$.

1.b. La grille 2 ne peut pas avoir de remplissage valide car le nombre 6 figure trois fois dans la grille ; or, ce nombre n'admet que deux décompositions distinctes : $6 = 1 + 5 = 2 + 4$.

1.c. Il s'agit d'une grille où les nombres sont décomposés en une somme de cinq nombres au plus. Or, le nombre le plus grand que l'on peut décomposer en somme de cinq nombres distincts parmi ceux de 1 à 9 est le nombre : $9 + 8 + 7 + 6 + 5 = 35$.

On ne peut donc pas remplacer l'un des nombres d'une case grisée par 38 qui est plus grand que 35.

2.a. $22 = 9 + 8 + 5 = 9 + 7 + 6$

2.b. $28 = 9 + 8 + 7 + 4 = 9 + 8 + 6 + 5$

2.c. $11 = 5 + 3 + 2 + 1$

2.d et 3.

Tableau de la question 2.d

	28↓	18↓		11↓	22↓
6→	4	2	6→	1	5
8→	7	1	10↓→	2	8
31→	8	7	4	3	9
28→	9	8	6	5	

Tableau de la question 3

	27↓	19↓		14↓	21↓
5→	3	2	16→	7	9
8→	7	1	8↓ ; 5→	1	4
29→	9	7	3	2	8
26→	8	9	5	4	

3. On utilise notamment les décompositions :

- De 5, 8 et 16 en somme de deux nombres : $\begin{cases} 5 = 1 + 4 = 2 + 3 \\ 8 = 1 + 7 = 2 + 6 = 3 + 5. \\ 16 = 7 + 9 \end{cases}$
- De 21 en somme de trois nombres : $21 = 9 + 8 + 4 = 9 + 7 + 5 = 8 + 7 + 6$

- De 14 et 27 en somme de quatre nombres :

  $14 = 1 + 2 + 3 + 8 = 1 + 2 + 4 + 7 = 1 + 2 + 5 + 6 = 1 + 3 + 4 + 6 = 2 + 3 + 4 + 5$

  $27 = 3 + 7 + 8 + 9 = 4 + 6 + 8 + 9 = 5 + 6 + 7 + 9$

Partie B

4. … avec Python

```
>>> def decomp():
        for a in range(3,16):
                for b in range(2,a):
                        for c in range(1,b):
                                if a+b+c==15:
                                        print({a,b,c})
```

```
>>> decomp()
{4, 5, 6}
{3, 5, 7}
{2, 6, 7}
{8, 3, 4}
{8, 2, 5}
{8, 1, 6}
{9, 2, 4}
{9, 5, 1}
{2, 10, 3}
{1, 10, 4}
{3, 1, 11}
{1, 2, 12}
```

5. Supposons que *n* soit un nombre impair au moins égal à 3 : $n = 2k + 1$ avec *k* entier strictement positif.

Pour tout entier *j* tel que $0 \leq j \leq k - 1,\ n = 2k + 1 = (k - j) + (k + 1 + j)$.

Nous obtenons ainsi *k* décompositions additives de *n* en somme de deux entiers strictement positifs distincts.

Or, si $n = 2k + 1$, alors $k = \frac{n-1}{2}$.

Supposons que *n* soit un nombre pair au moins égal à 4 : $n = 2k$ avec *k* entier ≥ 2.

Pour tout entier *j* tel que $1 \leq j \leq k - 1,\ n = 2k = (k - j) + (k + j)$.

Nous obtenons ainsi $(k - 1)$ décompositions additives de *n* en somme de deux entiers strictement positifs distincts. Or, si $n = 2k$, alors $k - 1 = \frac{n}{2} - 1$.

On note que les formules obtenues restent valables si *n* est égal à 1 ou à 2, donc finalement elles sont utilisables pour tout entier *n* strictement positif.

En conclusion, si *n* est impair (respectivement pair), on peut le décomposer de $\frac{n-1}{2}$ façons (respectivement de $\frac{n}{2} - 1$ façons) en somme de deux entiers strictement positifs distincts.