MINISTÈRE
DE L'ÉDUCATION
NATIONALE
ET DE LA JEUNESSE

Liberté
Égalité
Fraternité

Olympiades nationales de mathématiques 2023

Exercices Académiques, Grenoble

Les calculatrices sont autorisées selon la réglementation en vigueur.

L'épreuve se déroule par groupes de 1 à 4 élèves, chaque groupe rédige une seule copie, sans limitation du nombre de pages.

Il est conseillé aux groupes de candidats qui ne pourraient formuler une réponse complète à une question d'exposer le bilan des initiatives qu'ils ont pu prendre.

Les énoncés doivent être rendus au moment de quitter définitivement la salle de composition, ils pourront être restitués aux candidats le lendemain.

Chaque groupe de candidats traite **deux exercices.**

- Les candidats de la voie générale ayant suivi l'enseignement de spécialité de mathématiques doivent traiter les exercices 1 et 2.
- Les autres candidats doivent traiter les exercices 1 et 3.

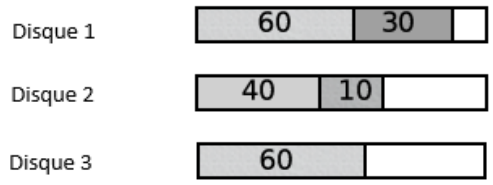
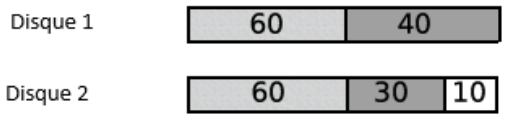
Exercice académique 1 (à traiter par tous les candidats)

Stockage de fichiers

On s'intéresse dans cet exercice au nombre minimal de disques durs à utiliser pour l'enregistrement de fichiers, les disques durs ayant tous une capacité de stockage identique.

Par exemple, on considère cinq fichiers de poids (ou taille) respectifs 60 Go, 30 Go, 40 Go, 10 Go et 60 Go. On souhaite enregistrer ces fichiers sur des disques durs de même capacité $c = 100$ Go.

On propose ci-dessous deux exemples de répartition, celle de droite étant optimale :

Avec trois disques durs	Avec deux disques durs
	

Partie A. Un nombre minimal de disques durs à utiliser.

1. On considère un ensemble de 7 fichiers de poids respectifs en Go : $10, 25, 40, 50, 60, 75, 90$.
   a) Proposer une répartition sur $n = 4$ disques durs de même capacité $c = 100$ Go.
   b) Pourquoi ne peut-on pas utiliser moins de 4 disques durs ?
2. Soit un ensemble de fichiers, dont la somme des poids (poids total) est P, à enregistrer sur n disques durs de capacité c. Montrer que l'on a toujours :

$$\frac{P}{c} \leqslant n$$

Pour toute la suite, on souhaite enregistrer k fichiers, notés F_0, F_1, …,F_{k-1} ($k \geq 1$) de poids respectifs $P_0, P_1, \ldots, P_{k-1}$, dans des disques durs de capacité c.

Les poids et la capacité seront exprimés en Go.

On appellera poids total P la somme des poids des fichiers : $P = P_0 + \cdots + P_{k-1}$.

On écrira les poids des fichiers à enregistrer sous la forme d'une liste : Liste_poids = [P_0,P_1,…,P_{k-1}].

On suppose que pour tout i entre 0 et $k-1$, on a : $0 < P_i \leq c$.

Partie B. Une première méthode de stockage : le stockage séquentiel.

La méthode de stockage séquentiel, expliquée ci-dessous, traite les fichiers dans l'ordre : F_0 puis F_1 etc.

On ouvre le premier disque dur.
Pour chaque fichier,
    s'il reste assez de place pour enregistrer ce fichier sur le disque dur ouvert, on l'enregistre sur ce disque ;
    sinon, on ferme définitivement le disque dur, on ouvre un nouveau disque dur et on enregistre le fichier sur le nouveau disque dur.

On souhaite écrire plus explicitement cette méthode sous la forme d'un algorithme.

Recopier et compléter sur la copie les lignes 7 et 10 de l'algorithme écrit en langage naturel ci-après correspondant à la fonction `sto_seq` :

- celle-ci prend en arguments la capacité commune c des disques durs et une liste d'entiers Liste_poids représentant les poids des fichiers à enregistrer ;
- elle renvoie le nombre de disques durs utilisés ainsi que, sous la forme d'une liste notée R dans l'algorithme, les espaces utilisés dans chacun de ces disques.

Exemple : l'appel `sto_seq(100, [60,30,40,10,60])` doit renvoyer `3, [90,50,60]`

Convention : si `L` est une liste, `L[0]` désigne le premier élément de la liste et `L[i]` désigne l'élément d'indice `i` (c'est donc le `i`+1-ième élément de la liste).

Numéro de lignes	Algorithme `sto_seq`
1	`sto_seq (c , Liste_poids)`
2	`    R ← liste vide`
3	`    i←0`
4	`    R[0] ← Liste_poids[0]`
5	`    j←1`
6	`    Tant que j < nombre d'éléments de Liste_poids:`
7	`        Si` ***à compléter*** `≤ c:`
8	`            R[i]← Liste_poids[j] + R[i]`
9	`        Sinon:`
10	`            R[i+1]←` ***à compléter***
11	`            i←i+1`
12	`        j←j+1`
13	`    renvoyer nombre d'éléments de R, R`

Partie C. Une deuxième méthode pour utiliser moins de disques durs.

On cherche ici à montrer que le nombre minimal n de disques durs utilisés pour enregistrer des fichiers dont le poids total est P vérifie :

$$n < 2\frac{P}{c}$$

1. On suppose dans cette question que tous les fichiers à enregistrer ont un poids strictement supérieur à $\frac{c}{2}$.
   En remarquant que, dans ce cas, le nombre de disques durs à utiliser est exactement égal au nombre de fichiers à enregistrer, montrer qu'on a : $n < 2\frac{P}{c}$.
2. On suppose ici que tous les fichiers sont de poids strictement supérieur à $\frac{c}{2}$ sauf un, de poids inférieur ou égal à $\frac{c}{2}$ et que celui-ci ne peut pas être regroupé avec les autres. Dans ce cas, le nombre de disques durs à utiliser est exactement égal au nombre de fichiers à enregistrer : on a donc n fichiers à enregistrer.

Quitte à changer l'ordre des fichiers, on peut écrire :

$$P_0 \leq \frac{c}{2} \text{ et } \frac{c}{2} < P_1 \leq \cdots \leq P_{n-1}.$$

a) Justifier que $c - P_1 < P_0$

b) Montrer que : $P_0 + (n-1)P_1 \leq P$.

c) En déduire que, dans ce cas, on a : $n < 2\frac{P}{c}$

3. On souhaite prouver le résultat dans le cas général.

   En regroupant de façon judicieuse les fichiers à enregistrer pour se ramener aux cas étudiés dans les questions précédentes, montrer que le nombre minimal de disques durs à utiliser pour stocker des fichiers dont le poids total est P, vérifie toujours :

$$n < 2\frac{P}{c}$$

Exercice académique 2 (à traiter par les candidats de voie générale ayant choisi la spécialité mathématiques)

La course

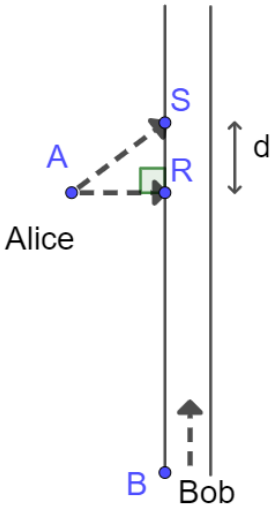
Sur un circuit automobile, une portion de piste rectiligne est protégée par une glissière de sécurité. Alice et Bob jouent au jeu suivant :

- Alice se trouve au point A à gauche de la piste, protégée par une glissière de sécurité, tandis que Bob se trouve sur la piste au niveau du point B.
- Bob est à moto et tente d'arriver au même niveau qu'Alice avant que celle-ci n'atteigne la glissière de sécurité.
- Si Alice parvient à atteindre la glissière au même moment que Bob ou avant lui, elle gagne 10 points, auxquels s'ajoutent autant de points que la distance d en mètres entre le point où elle a atteint la glissière et le point R, projeté orthogonal de A sur la droite modélisant la glissière.

Bob est située à 360 mètres du point R, c'est-à-dire BR = 360 m.
Alice est à 120 mètres de la glissière, c'est-à-dire AR = 120 m.

Partie 1 : bon départ !

Dans cette première partie, Alice et Bob partent en même temps (en ligne droite).
Alice court à 8 mètres par seconde mais Bob, à moto, avance trois fois plus vite (on néglige le temps d'accélération) !



1. Alice s'élance perpendiculairement à la glissière : va-t-elle réussir à atteindre la glissière sans se faire rattraper (et donc gagner 10 points) ?
2. Alice pourra-t-elle gagner 110 points ?
3. Combien de points peut-elle gagner au maximum ?

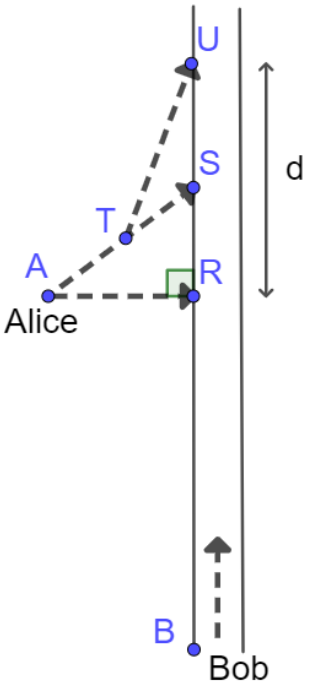
Partie 2 : mauvais départ !

Alice court toujours à 8 mètres par seconde et Bob, une fois la moto démarrée, avance trois fois plus vite qu'elle.

Mais, dans cette deuxième partie, Bob perd dix secondes à démarrer sa moto avant de se lancer à la poursuite d'Alice.

Alice s'en rend compte et décide, au moment où Bob démarre, de changer de direction (toujours en ligne droite) pour gagner davantage de points !

La course d'Alice est schématisée ci-contre par les segments [AT] et [TU], T représentant le point où se trouve Alice lorsque la moto démarre.



4. Combien de points peut-elle gagner au maximum ?

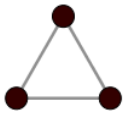
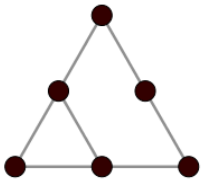
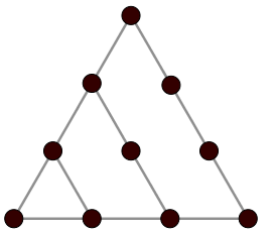
Exercice académique 3 (à traiter par les candidats n'ayant pas suivi la spécialité de mathématiques de voie générale)

Les nombres figurés

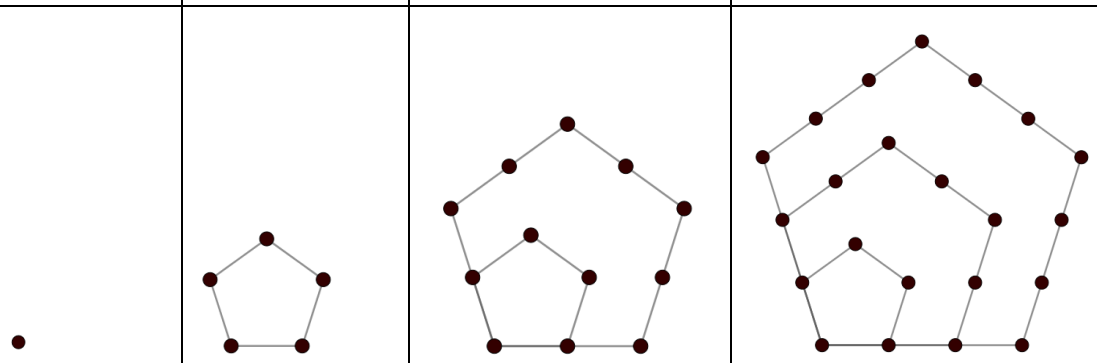
Un nombre figuré est un nombre entier pouvant être représenté par un ensemble de disques disposés de façon à former une figure géométrique comme un triangle, un rectangle, un pentagone ou un hexagone. Leur étude remonte à la Grèce antique et aurait été initiée par Pythagore et ses disciples.

On propose ici l'étude de certaines des propriétés des nombres triangulaires et des nombres pentagonaux.

On notera T_n le n ième nombre triangulaire égal au nombre de disques que contient un amas triangulaire régulier dont la base est formée de n disques.

$n = 1$	$n = 2$	$n = 3$	$n = 4$
			
$T_1 = 1$	$T_2 = 3$	$T_3 = 6$	$T_4 = 10$

On notera P_n le $n^{\text{ième}}$ nombre pentagonal égal au nombre de disques que contient un amas pentagonal régulier dont la base est formée de n disques (voir ci-après).

$n = 1$	$n = 2$	$n = 3$	$n = 4$
			
$P_1 = 1$	$P_2 = 5$	$P_3 = 12$	$P_4 = 22$

Les parties B et C sont indépendantes entre elles, mais nécessitent l'utilisation de résultats donnés dans la partie A.

Partie A : étude des nombres triangulaires et pentagonaux.

1) Dessiner sur la copie la représentation géométrique des nombres triangulaires des étapes 5 et 6 puis donner T_5 et T_6.
2) a) Dessiner sur la copie la représentation géométrique du nombre pentagonal de l'étape 5.
   b) Justifier que $P_5 = T_5 + 2\,T_4$ en coloriant, sur votre copie, de façon judicieuse les disques de la représentation géométrique de la question précédente.
3) On pose : $T_0 = 0$. On admet que, pour tout entier naturel n non nul, $P_n = T_n + 2\,T_{n-1}$
   a) Justifier que $T_n = \frac{n(n+1)}{2}$ pour tout entier naturel n non nul.
   b) Démontrer que $P_n = \frac{1}{2}n(3n-1)$ pour tout entier naturel n non nul.
   c) Montrer que pour tout entier naturel n non nul, on a : $P_n = \frac{1}{3}T_{3n-1}$.
4) On donne ci-dessous le code de la fonction Python `Triang` qui prend en paramètre un entier naturel non nul `n` et renvoie le nombre triangulaire égal à T_n.

Numéro de lignes	Code de la fonction `Triang`
1	`def Triang(n):`
2	`    T = 0.5 * n * (n+1)`
3	`    return T`

Écrire sur la copie le code Python d'une fonction `Penta` qui prend en paramètre un entier naturel non nul n et renvoie le nombre pentagonal P_n.

Partie B : écriture d'un nombre entier naturel non nul comme somme de nombres triangulaires ou pentagonaux.

On admet que tout entier naturel non nul peut s'écrire comme la somme d'au plus trois nombres triangulaires (pas nécessairement distincts).

1) a) Vérifier que $220 = T_4 + T_{14} + T_{14}$.
   b) Proposer une autre écriture de 220 comme somme de trois nombres triangulaires.

On peut montrer également que tout entier naturel non nul peut s'écrire comme la somme d'au plus cinq nombres pentagonaux (pas nécessairement distincts)..

2) a) Vérifier que $220 = P_2 + P_3 + P_5 + P_6 + P_9$.
   b) Proposer une écriture de 220 comme somme de trois nombres pentagonaux.

Partie C. Une méthode pour savoir si un entier naturel non nul est un nombre pentagonal ou non.

Soit N un nombre entier naturel non nul. On considère l'équation (E) : $3x^2 - x - 2N = 0$ d'inconnue x.

1) À l'aide de la question A.3) b), montrer que N est un nombre pentagonal si et seulement s'il existe un entier naturel non nul n solution de l'équation (E).
2) Vérifier que $\frac{1+\sqrt{24N+1}}{6}$ est une solution de (E).

On admettra pour la suite que $\frac{1+\sqrt{24N+1}}{6}$ est la seule solution strictement positive de (E).

3) En déduire que 330 est un nombre pentagonal et donner le nombre de disques que contient sa base. L'entier 329 est-il un nombre pentagonal ?
4) L'algorithme 1 ci-dessous, écrit en langage naturel, permet de déterminer si un entier naturel non nul est un nombre pentagonal. Il doit afficher VRAI si le nombre est un nombre pentagonal et FAUX sinon.
   Recopier et compléter sur la copie les lignes 3 et 4.

Numéro de lignes	Algorithme 1
1	N ← nombre entier non nul à tester
2	si $\frac{1+\sqrt{24N+1}}{6}$ est un nombre entier
3	alors afficher « ***à compléter*** »
4	sinon afficher « ***à compléter*** »