

Problème 2025

Paul Jolissaint

1. Voici les premières décompositions :

$$\begin{array}{lll} 5 = 2^2 + 1^2 & 6 = 2^2 + 1^2 + 1^2 & 7 = 2^2 + 1^2 + 1^2 + 1^2 \\ 8 = 2^2 + 2^2 & 9 = 3^2 & 10 = 3^2 + 1^2 \\ 11 = 3^2 + 1^2 + 1^2 & 12 = 3^2 + 1^2 + 1^2 + 1^2 & 13 = 3^2 + 2^2 \\ 14 = 3^2 + 2^2 + 1^2 & 15 = 3^2 + 2^2 + 1^2 + 1^2 & 16 = 4^2 \\ 17 = 4^2 + 1^2 & 18 = 4^2 + 1^2 + 1^2 & 19 = 4^2 + 1^2 + 1^2 + 1^2 \\ 20 = 4^2 + 2^2 & 21 = 4^2 + 2^2 + 1^2 & 22 = 4^2 + 2^2 + 1^2 + 1^2 \\ 23 = 3^2 + 3^2 + 2^2 + 1^2 & 24 = 4^2 + 2^2 + 2^2 & 25 = 5^2. \end{array}$$

2. Non : par exemple, $25 = 5^2 = 4^2 + 2^2 + 2^2 + 1^2$.
3. Tout entier positif est somme d'au plus quatre carrés (cf. ci-dessous).
4. Oui : $58 = 7^2 + 3^2$.
5. Non, par inspection et par le critère ci-dessous, $21 = 3 \cdot 7$, où 3 et 7 sont de la forme $4k + 3$.
6. Oui, et on connaît une condition nécessaire et suffisante pour que ce soit le cas : l'entier positif n est somme de deux carrés si et seulement si, dans sa décomposition en facteurs premiers, tous ceux qui sont de la forme $4k + 3$ apparaissent à une puissance paire.
Par exemple, $58 = 2 \cdot 29$ et $29 = 4 \cdot 7 + 1$, ou encore

$$1125 = 30^2 + 15^2 = 3^2 \cdot 5^3.$$

7. L'identité de Lagrange se vérifie par calcul direct. Observons que, en posant $z = a + ib$ et $w = c + id$ (nombres complexes), cette identité est exactement la multiplicativité du module des nombres complexes :

$$(a^2 + b^2)(c^2 + d^2) = |z|^2|w|^2 = |zw|^2 = (ac + bd)^2 + (ad - bc)^2$$

$$\text{car } zw = (a + ib)(c + id) = (ac - bd) + i(ad + bc).$$

8. $18 \cdot 58 = (3^2 + 3^2)(3^2 + 7^2) = (3 \cdot 3 + 3 \cdot 7)^2 + (3 \cdot 7 - 3 \cdot 3)^2 = 30^2 + 12^2 = 1044$.
9. Si $n = a^2 + b^2$, comme $2 = 1^2 + 1^2$, on obtient

$$2n = (a + b)^2 + (a - b)^2.$$

10. Soit $2n = a^2 + b^2$. Comme la parité d'un entier est la même que celle de son carré, a et b sont soit tous les deux pairs, soit tous les deux impairs. Par suite, les nombres $a \pm b$ sont toujours pairs, et on vérifie immédiatement que

$$n = \left(\frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2.$$

11. Le nombre $1344 = 2^6 \cdot 21$ n'est pas somme de deux carrés car s'il l'était, en le divisant 6 fois par 2, on trouverait que 21 est somme de deux carrés, ce qui n'est pas le cas.
12. Oui : Par exemple, la suite $(2^k)_{k \geq 1}$ par le point 10.
13. Oui : par exemple la suite $(2^k \cdot 21)_{k \geq 1}$.

Une preuve du théorème de Lagrange

On rappelle que, pour tout entier $m > 0$, la notation $x \equiv y \pmod{m}$ signifie que m divise la différence $x - y$.

Lemme 1 *Soit p un nombre premier impair. Alors il existe deux entiers*

$$x, y \in E := \left\{ 0, 1, 2, \dots, \frac{p-1}{2} \right\}$$

tels que $1 + x^2 + y^2 = kp$ avec k entier tel que $0 < k < p$.

PREUVE. Tout d'abord, si $x, y \in E$ sont tels que $x \neq y$ alors x^2 et y^2 sont distincts \pmod{p} . En effet, si p divise $x^2 - y^2$, alors, p divise $(x - y)(x + y)$, et comme il est premier, il en divise un des deux. Par exemple, s'il divise $x - y$, quitte à les permuter, on suppose sans perte que $x > y$, et alors il existe un entier positif k tel que $x - y = kp$. Or, on aurait

$$kp = x - y \leq x + y \leq 2 \cdot \frac{p-1}{2} = p - 1,$$

ce qui est une contradiction. De même si p divise $x + y$. Ainsi, les $\frac{p-1}{2} + 1 = \frac{p+1}{2}$ nombres de E fournissent $\frac{p+1}{2}$ valeurs distinctes de $x^2 \pmod{p}$. De même, ces éléments fournissent $\frac{p+1}{2}$ valeurs distinctes de $-1 - y^2 \pmod{p}$ (c'est la même preuve que ci-dessus). Comme il y a au plus p valeurs distinctes \pmod{p} , il existe $x, y \in E$ tels que $x^2 \equiv -1 - y^2 \pmod{p}$, autrement dit, $1 + x^2 + y^2 \equiv 0 \pmod{p}$. Donc $1 + x^2 + y^2 = kp$ pour un entier positif k . Or, comme $x, y \leq (p-1)/2$, on a

$$kp = 1 + x^2 + y^2 < 1 + 2 \cdot \frac{p^2}{4} = 1 + \frac{p^2}{2}$$

ce qui implique que $k < p$. En effet, sinon, on aurait $p^2 \leq kp < 1 + p^2/2$ et donc $p^2/2 < 1$, ce qui est impossible. \square

La preuve ci-dessous, assez classique, est fortement inspirée de : D. Duvernay, *Théorie des nombres*, éd. Dunod, Paris, 1998.

Théorème 2 (Lagrange) *Tout entier positif est somme d'au plus quatre carrés d'entiers.*

PREUVE. Comme dans le cas de l'identité de Lagrange (point 7 ci-dessus), il existe une identité (qui se vérifie directement et qui admet la même interprétation que celle de Lagrange, mais pour les quaternions, qui sont les nombres de la forme $a+bi+cj+dk$ avec $a, b, c, d \in \mathbb{R}$ et où i, j, k satisfont des règles du type $i^2 = j^2 = k^2 = -1$, $ij = -ji = k$ etc.) :

$$(x^2 + y^2 + z^2 + w^2)(x'^2 + y'^2 + z'^2 + w'^2) = \\ (xx' + yy' + zz' + ww')^2 + (xy' - yx' + wz' - zw')^2 \\ + (xz' - zx' + yw' - wy')^2 + (xw' - wx' + zy' - yz')^2. \quad (1)$$

Ainsi le produit de deux sommes de quatre carrés est encore une somme de quatre carrés, et puisque $2 = 1^2 + 1^2 + 0^2 + 0^2$, pour démontrer le théorème, en utilisant le fait que tout entier positif est un produit de nombre premiers, il suffit de démontrer que tout nombre premier impair est une somme de quatre carrés.

Soit donc $p \geq 3$ un nombre premier impair. Par le lemme 1, on choisit deux entiers x, y tels que $|x|, |y| < p/2$ et que $1 + x^2 + y^2 = kp$. Soit alors $0 < m \leq k$ le plus petit entier positif tel qu'il existe des entiers a, b, c, d tels que

$$a^2 + b^2 + c^2 + d^2 = mp$$

avec $m < p$. On doit montrer qu'en fait $m = 1$.

Affirmation m est impair.

En effet, si m était pair, alors 0, 2 ou 4 des nombres a, b, c, d seraient impairs, par exemple a et b dans le cas de 2. Alors les nombres $a + b, a - b, c + d$ et $c - d$ seraient tous pairs et on aurait

$$\left(\frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2 + \left(\frac{c+d}{2}\right)^2 + \left(\frac{c-d}{2}\right)^2 = \frac{m}{2} \cdot p$$

ce qui contredirait la minimalité de m .

Ainsi, m est impair, et supposons par l'absurde que $m \geq 3$. Soient alors des représentants a', b', c', d' de $a, b, c, d \pmod{m}$ respectivement tels que $|a'|, |b'|, |c'|, |d'| < m/2$. (Donc $a' \equiv a \pmod{m}$ etc.)

Posons $n = a'^2 + b'^2 + c'^2 + d'^2$, de sorte que $n \equiv a^2 + b^2 + c^2 + d^2 = mp \equiv 0 \pmod{m}$. On a $n > 0$ sinon $a' = b' = c' = d' = 0$ et il existerait des entiers $\alpha, \beta, \gamma, \delta$ tels que $a = \alpha m, b = \beta m$ etc. Donc

$$mp = a^2 + b^2 + c^2 + d^2 = (\alpha^2 + \beta^2 + \gamma^2 + \delta^2)m^2$$

ce qui impliquerait que $m = p$, mais cela contredit l'hypothèse que $m < p$. Donc $n > 0$.

Mais $n < 4(m^2/4) = m^2$, donc il existe un entier $0 < u < m$ tel que $n = um = a'^2 + b'^2 + c'^2 + d'^2$ et $mp = a^2 + b^2 + c^2 + d^2$. Par l'identité (1), il existe des entiers A, B, C, D tels que

$$um^2p = (um)(mp) = A^2 + B^2 + C^2 + D^2.$$

En inspectant (1), on voit que $A \equiv B \equiv C \equiv D \equiv 0 \pmod{m}$ par définition de a', b', \dots . Ainsi, $up = A'^2 + B'^2 + C'^2 + D'^2$ après simplification par m^2 , avec $0 < u < m$, ce qui contredit la minimalité de m . Donc $m = 1$ et p est somme de quatre carrés. \square

Problème du cercle de Mathématiques et Physique 2025

1. **Écrire une décomposition en somme de carrés de chacun des nombres 5 et 22.**

$$5 = 2^2 + 1^2 \quad \text{et} \quad 22 = 3^2 + 3^2 + 2^2$$

2. **Ces décompositions sont-elles les seules sommes de carrés égales à 5 ou à 22 ?**

Non, par exemple $5 = 1^2 + 1^2 + 1^2 + 1^2 + 1^2$ et $22 = 1^2 + 2^2 + 2^2 + 2^2 + 3^2$

3. **Tout entier naturel non nul est-il décomposable en somme de carrés ?**

Oui, par exemple $n = \underbrace{1^2 + 1^2 + \dots + 1^2}_{n \text{ fois}}$

4. **Le nombre 58 est-il un nombre bicarré ?**

Oui : $58 = 3^2 + 7^2$

5. **Le nombre 21 est-il un nombre bicarré ?**

Non, car $21 = 4^2 + 5 = 3^2 + 12 = 2^2 + 17 = 1^2 + 20 = 0^2 + 21$ et aucun des nombres grisés n'est un carré parfait.

6. **Y a-t-il une infinité de nombres bicarrés ?**

Oui. Soient a et $b \in \mathbb{N}$. Le nombre $n = a^2 + b^2$ est un nombre bicarré. Et comme l'ensemble \mathbb{N} possède une infinité d'éléments, il y a une infinité de nombres bicarrés.

7. **Commencer par montrer l'égalité de Lagrange, valable pour tous nombres a, b, c et d tels que :**

$$(a^2 + b^2)(c^2 + d^2) = (ac + bd)^2 + (ad - bc)^2$$

Egalité vraie !

8. **Calculer une décomposition en somme de deux carrés de 18×58 .**

$$18 \times 58 = (3^2 + 3^2)(3^2 + 7^2) = (9 + 21)^2 + (21 - 9)^2 = 30^2 + 12^2$$

9. **En déduire une décomposition en somme de deux carrés du double d'un nombre bicarré.**

$$n = a^2 + b^2 \rightarrow 2n = 2(a^2 + b^2) = (1^2 + 1^2)(a^2 + b^2) = (a + b)^2 + (b - a)^2$$

Exemple : $n = 12^2 + 7^2 = 193$

$$2n = 2(12^2 + 7^2) = (1^2 + 1^2)(12^2 + 7^2) = (12 + 7)^2 + (12 - 7)^2 = 19^2 + 5^2 = 386$$

10. **Montrer que la moitié d'un nombre pair bicarré est un nombre bicarré.**

Si n est pair et bicarré, alors a et b sont de même parité.

- a et b sont pairs $\rightarrow a = 2p$ et $b = 2q \rightarrow \frac{n}{2} = \frac{(2p)^2 + (2q)^2}{2} = 2(p^2 + q^2) = (p + q)^2 + (p - q)^2$

- a et b sont impairs $\rightarrow a = 2p+1$ et $b = 2q+1 \rightarrow \frac{n}{2} = \frac{(2p+1)^2 + (2q+1)^2}{2}$
 $= 2p^2 + 2p + 2q^2 + 2q + 1$
 $= (p + q + 1)^2 + (p - q)^2$

Exemple : $p = 9$ et $q = 17 \rightarrow n = (2 \times 9 + 1)^2 + (2 \times 17 + 1)^2 = 1586$

$$\frac{n}{2} = (9 + 17 + 1)^2 + (9 - 17)^2 = 793$$

11. **1344 est-il un nombre bicarré ?**

$1344 = 2^6 \times 21$. Comme 21 n'est pas bicarré, 1344 n'est pas bicarré.

12. **Existe-t-il une infinité de nombres pairs bicarrés ?**

Oui. Soient a et $b \in \mathbb{N}$ et de même parité. Le nombre $n = a^2 + b^2$ est un nombre bicarré pair. Et comme l'ensemble \mathbb{N} possède une infinité d'éléments, il y a une infinité de nombres bicarrés pairs.

13. **Existe-t-il une infinité de nombres pairs qui ne sont pas bicarrés ?**

Oui, par exemple ceux de la forme $n = 2 \times 3^{2k+1}$ ($k \geq 0$). Exemple : 6 ; 54 ; 486 ; ...

Essai de résolution du Problème de Mathématiques et Physique 2025

1. $5 = 2^2 + 1^2$
 $22 = 4^2 + 2^2 + 1^2 + 1^2$
2. Non, pour chaque valeur, on peut, par exemple, faire une somme de plusieurs 1^2 .
3. Oui, cf. démonstration de Lagrange 1770, énoncée au numéro 13.
4. 58 est un nombre bicarré: $58 = 49 + 9 = 7^2 + 3^2$
5. 21 n'est pas un nombre bicarré: $21 = 16 + 5 = 4^2 + 5$
ou $21 = 9 + 13$ ou $21 = 4 + 17$ ou $21 = 1 + 20$
6. Oui, il suffit de prendre une des deux valeurs au carré égale à 1 n'importe quelle valeur pour le second nombre à élever au carré: $1^2 + s^2 \quad s \in \square$
7.
$$(a^2 + b^2)(c^2 + d^2) = a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2 + 2abcd - 2abcd =$$
$$= a^2c^2 + 2abcd + b^2d^2 + a^2d^2 - 2abcd + b^2c^2 = (ac + bd)^2 + (ad - bc)^2$$
8. $18 = 3^2 + 3^2 \quad a = 3 \quad \text{et} \quad b = 3$
 $58 = 7^2 + 3^2 \quad c = 7 \quad \text{et} \quad d = 3$

En appliquant l'identité de Lagrange (cf. numéro 7), on obtient:

$$18 \times 58 = (3^2 + 3^2)(7^2 + 3^2) = (3 \cdot 7 + 3 \cdot 3)^2 + (3 \cdot 3 - 3 \cdot 7)^2 = 30^2 + (-12)^2$$

$$\text{On a bien: } 18 \times 58 = 1044 \quad \text{et} \quad 30^2 + 12^2 = 900 + 144 = 1044$$

9. Soit n un nombre bicarré. Donc, $n = x^2 + y^2$
On prend le double de n , donc $2n = 2x^2 + 2y^2$
On applique l'égalité de Lagrange:

$$2 = a^2 + b^2 = 1^2 + 1^2 \quad \text{et} \quad a = 1 \quad \text{et} \quad b = 1$$

$$n = x^2 + y^2 \quad \text{ou} \quad n = c^2 + d^2 \quad \text{avec} \quad c = x \quad \text{et} \quad d = y$$

$$\text{Lagrange:} \quad 2n = (1^2 + 1^2)(c^2 + d^2) = (1 \cdot c + 1 \cdot d)^2 + (1 \cdot d - 1 \cdot c)^2 = (c + d)^2 + (d - c)^2$$

$$\text{Et donc:} \quad 2n = (x + y)^2 + (y - x)^2 \quad \text{Il s'agit bien d' nombre bicarré}$$

10. Soit n un nombre bicarré. Donc, $n = x^2 + y^2$

n étant pair, on peut écrire : $n = 2p$, p étant donc la moitié de n .

Il s'agit de démontrer que p est un nombre bicarré.

$$2p = n = (a^2 + b^2)(c^2 + d^2)$$

On peut écrire: $2 = (a^2 + b^2)$ et $p = (c^2 + d^2)$

Donc: $a = 1$ et $b = 1$

En appliquant Lagrange:

$$2p = (c + d)^2 + (d - c)^2 = c^2 + 2cd + d^2 + d^2 - 2dc + c^2 = 2c^2 + 2d^2 = 2(c^2 + d^2) \text{ et}$$

$p = c^2 + d^2$ ce qui est bien un nombre bicarré

11. On divise le nombre pair 1344 par 2, 6 fois et on obtient: $1344 = 2^6 \cdot 21 = 64 \cdot 21$

Si 1344 était un nombre pair bicarré, les 6 nombres obtenus par division par 2 seraient bicarrés, y compris le dernier nombre pair qui est 42 et la moitié de 43, donc 21 devrait aussi être bicarré. Mais, 21 n'est pas bicarré (cf. numéro 5). Donc, 1344 n'est pas bicarré.

12. Oui, si on a un nombre pair bicarré, tous les nombres obtenus par une multiplication par une puissance de 2 sont pairs et bicarrés.

13. Oui, si on multiplie par 2 un nombre impair et qui n'est pas un bicarré, on obtient un nombre pair, qui n'est pas un bicarré. (est-ce correct?).

14. Soit N le nombre à décomposer.

T: nombre entier de travail.

V: tableau d'entiers contenant les 4 carrés de la décomposition (indices 1 à 4).

D: nombre entier contenant provisoirement un des 4 carrés.

J'utilise **Racine()** pour la fonction racine carrée (le résultat est un réel) et la fonction **Entier()** pour prendre la partie entière d'un réel. La fonction **Chiffres après la virgule()** prend la valeur d'un réel après la virgule.

Initialisation:

T:=N;

V[1]:=Entier(Racine(N));

D:=V[1]+1; *pour compenser le -1 du début de bouclée

Boucles:

Répéter D:=D-1; *pour préparer la prochaine valeur du premier carré

V[1]:=D;

Pour i:=2

T:=T-V[i-1]²; *nouvelle valeur à décomposer

V[i]:=Entier(Racine(T)); *nouveau nombre à stocker

à i=4; *à faire pour i=2 à i=4

Jusqu'à Chiffres après la virgule (Racine(T))=0; *jusqu'à ce que la dernière valeur

*soit un entier

Affichage:

Afficher les 4 carrés contenus dans le Tableau.

P.S. Faute de temps (...) l'algorithme est fait à la va-vite et non testé par un programme.

Problème des Cercle de Mathématiques et Physique 2025

① Décomposition en somme de carrés de 5 et 22

$$5 = 2^2 + 1^2$$

$$22 = 3^2 + 3^2 + 2^2$$

② Il y a d'autres solutions

$$5 = 1^2 + 1^2 + 1^2 + 1^2 + 1^2$$

$$22 = 4^2 + 2^2 + 1^2 + 1^2$$

etc.

③ Tout entier naturel est décomposable en autant de carrés de 1^2

④ $58 = 7^2 + 3^2 \Rightarrow$ bicarré

⑤ 21 n'est pas bicarré

$$21 = 4^2 + 2^2 + 1^2$$

$$21 = 3^2 + 3^2 + 1^2 + 1^2 + 1^2$$

⑥ Il y a une infinité de nombres bicarrés

$$\textcircled{7} (a^2 + b^2)(c^2 + d^2) = a^2c^2 + a^2d^2 + b^2c^2 + b^2d^2$$

$$(ac + bd)^2 + (ad - bc)^2 =$$

$$= a^2c^2 + 2abcd + b^2d^2 + a^2d^2 - 2abcd + b^2c^2$$

$$= a^2c^2 + b^2d^2 + a^2d^2 + b^2c^2$$

$$\textcircled{8} 18 \times 58 = (3^2 + 3^2)(7^2 + 3^2) =$$

$$= (3 \cdot 7 + 3 \cdot 3)^2 + (3 \cdot 3 - 3 \cdot 7)^2$$

Lagrange

$$= \underline{\underline{30^2 + 12^2 = 1044}}$$

$$\textcircled{9} a^2 + b^2 = n$$

$$(a+b)^2 + (a-b)^2 = 2n$$

$$\textcircled{10} \text{ Hypothèse } a^2 + b^2 = n \text{ pair}$$

$\Rightarrow a^2$ et b^2 sont les deux soit pairs soit impairs

Si a^2 et b^2 sont pairs, alors ils sont également divisibles par 4 et $\frac{a}{2}$ et $\frac{b}{2}$ sont entiers

$$\Rightarrow \left(\frac{a}{2} + \frac{b}{2}\right)^2 + \left(\frac{a}{2} - \frac{b}{2}\right)^2 = \frac{n}{2}$$

Si a^2 et b^2 sont les deux impairs alors a et b sont impairs et $a+b$ et $a-b$ sont pairs

donc $\frac{a+b}{2}$ et $\frac{a-b}{2}$ sont entiers

$$\Rightarrow \left(\frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2 = \frac{n}{2}$$

$$\textcircled{11} \begin{aligned} 1344 &= 36^2 + 48 \\ &= 35^2 + 119 \\ &= 34^2 + 128 \\ &= 33^2 + 255 \\ &= 32^2 + 320 \\ &= 31^2 + 383 \end{aligned}$$

$$\begin{aligned}
1344 &= 30^2 + 444 \\
&= 29^2 + 503 \\
&= 28^2 + 560 \\
&= 27^2 + 615 \\
&= 26^2 + 668 \\
&= 25^2 + 719
\end{aligned}$$

ou vient de dépasser la moitié de 1344
 Réponse 1344 n'est pas un nombre bicarré (*)

(12) Il existe une infinité de nombres pairs bicarrés

par exemple tous les nombres de la forme

$$n = 2^m \cdot 1044 \quad \text{avec } m \in \mathbb{N}$$

(13) Il existe sûrement une infinité de nombres pairs qui ne sont pas bicarrés (les nombres bicarrés constituent plutôt l'exception)

si n est un nombre bicarré pair alors en général $n+2$ n'est pas bicarré

(*) On peut décomposer 1344 en 4 carrés

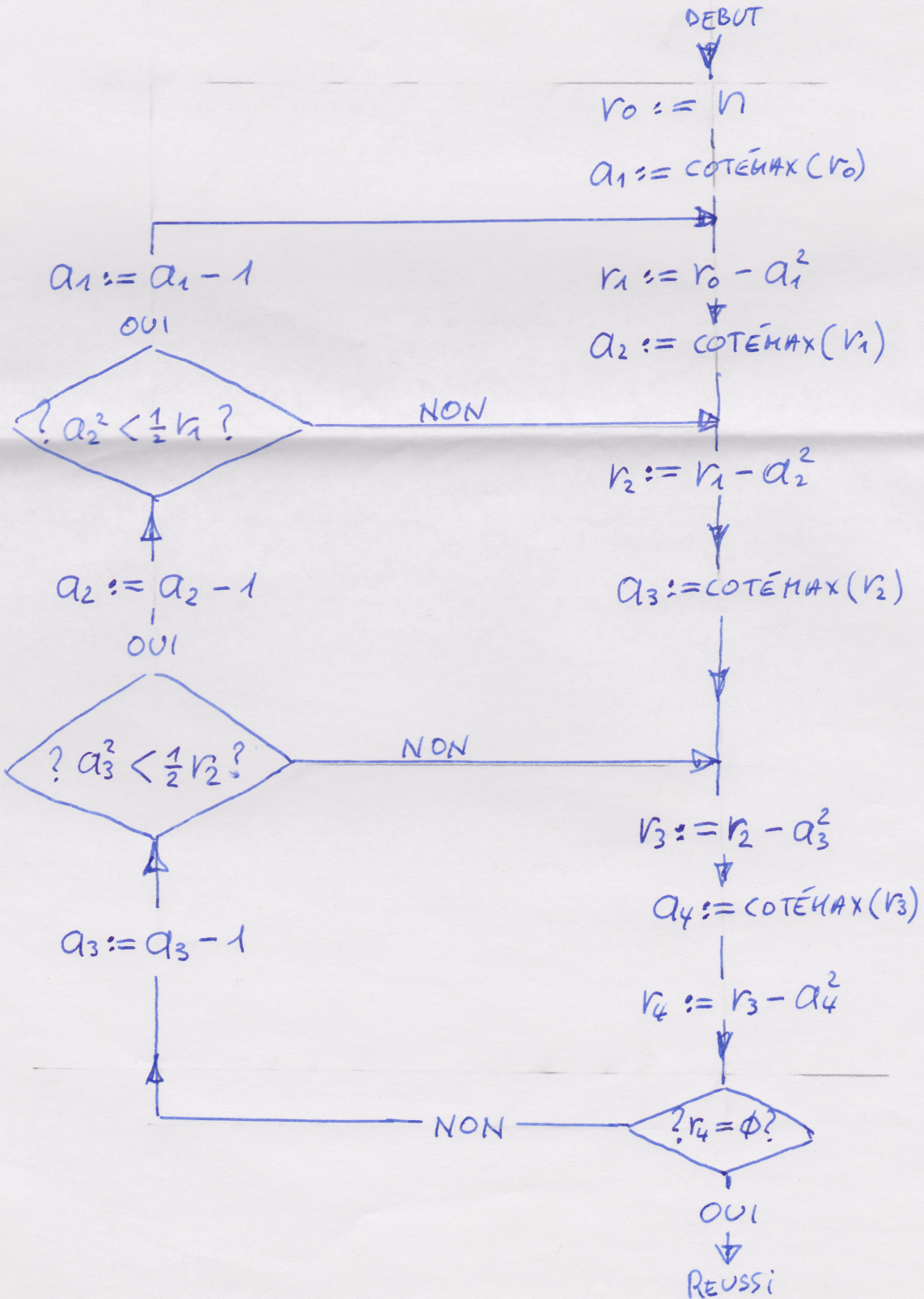
$$1344 = 20^2 + 20^2 + 20^2 + 12^2$$

14) Algorithme

Notations

$$n = a_1^2 + a_2^2 + a_3^2 + a_4^2 + r_4$$

COTÉMAX(n) = le côté du plus grand carré contenu dans n



foursquares.f

```

1  C  =====
2  C  Lagrange's Four Square Theorem - Pure Fortran 77
3  C  =====
4  C  Fixed-form source: statements start in column 7,
5  C  continuation lines have a character in column 6.
6  C  =====
7  PROGRAM FOURSQ
8  INTEGER N, A, B, C, D
9  INTEGER TNUM(6)
10 LOGICAL FOUND
11 INTEGER I, NTHUM
12 DATA TNUM / 7, 15, 23, 100, 123, 999 /
13
14 NTHUM = 6
15
16 PRINT *, '===== '
17 PRINT *, 'Lagrange's Four Square Theorem - Fortran 77'
18 PRINT *, '===== '
19 PRINT *, ' '
20
21 DO 100 I = 1, NTHUM
22     N = TNUM(I)
23     CALL FOURSUB(N, A, B, C, D, FOUND)
24     IF (FOUND) THEN
25         WRITE(*, 200) N, A, B, C, D,
26 &             A*A + B*B + C*C + D*D
27     ELSE
28         PRINT *, 'Could not find decomposition for ', N
29     END IF
30 100 CONTINUE
31
32 200 FORMAT('n = ', I4, ' = ', I4, '^2 + ', I4, '^2 + ',
33 &         I4, '^2 + ', I4, '^2 = ', I6)
34
35 STOP
36 END
37
38 C  -----
39 C  Subroutine: FOURSUB
40 C  Finds a, b, c, d such that n = a^2 + b^2 + c^2 + d^2
41 C  -----
42 SUBROUTINE FOURSUB(N, A, B, C, D, FOUND)
43 INTEGER N, A, B, C, D
44 LOGICAL FOUND
45 INTEGER SQRTN, REMAIN, TEMP
46 INTEGER MAXA, MAXB, MAXC
47 INTEGER ISQRT
48
49 FOUND = .FALSE.
50 A = 0
51 B = 0

```

```

52      C = 0
53      D = 0
54
55 C    Handle n = 0
56      IF (N .EQ. 0) THEN
57          FOUND = .TRUE.
58          RETURN
59      END IF
60
61 C    Check if n is a perfect square (1 square)
62      SQRTN = INT(SQRT(REAL(N)))
63      IF (SQRTN * SQRTN .EQ. N) THEN
64          A = SQRTN
65          FOUND = .TRUE.
66          RETURN
67      END IF
68
69 C    Check for sum of 2 squares
70      MAXA = INT(SQRT(REAL(N)))
71      DO 10 A = 1, MAXA
72          REMAIN = N - A * A
73          ISQRT = INT(SQRT(REAL(REMAIN)))
74          IF (ISQRT * ISQRT .EQ. REMAIN) THEN
75              B = ISQRT
76              FOUND = .TRUE.
77              RETURN
78          END IF
79      10 CONTINUE
80
81 C    Check for sum of 3 squares
82 C    Skip if n is of form 4^k(8m+7) -- needs at least 4 squares
83      TEMP = N
84      20 CONTINUE
85      IF (MOD(TEMP, 4) .EQ. 0) THEN
86          TEMP = TEMP / 4
87          GOTO 20
88      END IF
89
90      IF (MOD(TEMP, 8) .NE. 7) THEN
91          MAXA = INT(SQRT(REAL(N)))
92          DO 30 A = 1, MAXA
93              IF (N - A*A .LE. 0) GOTO 40
94              MAXB = INT(SQRT(REAL(N - A*A)))
95              DO 31 B = A, MAXB
96                  REMAIN = N - A*A - B*B
97                  IF (REMAIN .LT. 0) GOTO 30
98                  ISQRT = INT(SQRT(REAL(REMAIN)))
99                  IF (ISQRT * ISQRT .EQ. REMAIN .AND.
100      &                ISQRT .GE. B) THEN
101                      C = ISQRT
102                      FOUND = .TRUE.
103                      RETURN
104                  END IF
105      31 CONTINUE

```

```
106      30      CONTINUE
107      40      CONTINUE
108      END IF
109
110 C      Find sum of 4 squares (always possible by Lagrange's theorem)
111      MAXA = INT(SQRT-REAL(N))
112      DO 50 A = 1, MAXA
113          IF (N - A*A .LE. 0) GOTO 90
114          MAXB = INT(SQRT-REAL(N - A*A))
115          DO 51 B = A, MAXB
116              IF (N - A*A - B*B .LE. 0) GOTO 50
117              MAXC = INT(SQRT-REAL(N - A*A - B*B))
118              DO 52 C = B, MAXC
119                  REMAIN = N - A*A - B*B - C*C
120                  IF (REMAIN .LT. 0) GOTO 51
121                  ISQRT = INT(SQRT-REAL(REMAIN))
122                  IF (ISQRT * ISQRT .EQ. REMAIN .AND.
123                      &          ISQRT .GE. C) THEN
124                      D = ISQRT
125                      FOUND = .TRUE.
126                      RETURN
127                  END IF
128      52      CONTINUE
129      51      CONTINUE
130      50      CONTINUE
131      90      CONTINUE
132
133      RETURN
134      END
135
```

foursquares.f90

Claude Fuhrer Fortran 90

```

1  program four_squares_theorem
2      implicit none
3      integer :: n, a, b, c, d
4      logical :: found
5      integer, dimension(6) :: test_numbers = [7, 15, 23, 100, 123, 999]
6      integer :: i
7
8      print *, "======"
9      print *, "Lagrange's Four Square Theorem - Fortran 90"
10     print *, "======"
11     print *, ""
12
13     ! Run examples
14     do i = 1, size(test_numbers)
15         n = test_numbers(i)
16         call find_four_squares(n, a, b, c, d, found)
17
18         if (found) then
19             print '(A,I0,A,I0,A,I0,A,I0,A,I0,A,I0)', &
20                 "n = ", n, " = ", a, "² + ", b, "² + ", c, "² + ", d, &
21                 "² = ", a*a + b*b + c*c + d*d
22         else
23             print *, "Could not find decomposition for ", n
24         end if
25     end do
26
27 contains
28
29     subroutine find_four_squares(n, a, b, c, d, found)
30         integer, intent(in) :: n
31         integer, intent(out) :: a, b, c, d
32         logical, intent(out) :: found
33         integer :: sqrt_n, remainder, temp
34         integer :: max_a, max_b, max_c
35
36         found = .false.
37         a = 0
38         b = 0
39         c = 0
40         d = 0
41
42         ! Handle n = 0
43         if (n == 0) then
44             found = .true.
45             return
46         end if
47
48         ! Check if n is a perfect square (1 square)
49         sqrt_n = int(sqrt(real(n)))
50         if (sqrt_n * sqrt_n == n) then
51             a = sqrt_n

```

```

52         found = .true.
53         return
54     end if
55
56     ! Check for sum of 2 squares
57     max_a = int(sqrt(real(n)))
58     do a = 1, max_a
59         remainder = n - a * a
60         b = int(sqrt(real(remainder)))
61         if (b * b == remainder) then
62             found = .true.
63             return
64         end if
65     end do
66
67     ! Check for sum of 3 squares
68     ! Skip if n is of form 4^k(8m+7)
69     temp = n
70     do while (mod(temp, 4) == 0)
71         temp = temp / 4
72     end do
73
74     if (mod(temp, 8) /= 7) then
75         max_a = int(sqrt(real(n)))
76         do a = 1, max_a
77             if (n - a*a <= 0) exit
78             max_b = int(sqrt(real(n - a*a)))
79             do b = a, max_b
80                 remainder = n - a*a - b*b
81                 if (remainder < 0) exit
82                 c = int(sqrt(real(remainder)))
83                 if (c * c == remainder .and. c >= b) then
84                     found = .true.
85                     return
86                 end if
87             end do
88         end do
89     end if
90
91     ! Find sum of 4 squares
92     max_a = int(sqrt(real(n)))
93     do a = 1, max_a
94         if (n - a*a <= 0) exit
95         max_b = int(sqrt(real(n - a*a)))
96         do b = a, max_b
97             if (n - a*a - b*b <= 0) exit
98             max_c = int(sqrt(real(n - a*a - b*b)))
99             do c = b, max_c
100                remainder = n - a*a - b*b - c*c
101                if (remainder < 0) exit
102                d = int(sqrt(real(remainder)))
103                if (d * d == remainder .and. d >= c) then
104                    found = .true.
105                    return

```

```
106         end if
107     end do
108 end do
109 end do
110
111 end subroutine find_four_squares
112
113 end program four_squares_theorem
114
```

FourSquares.mod

Claude Fuhrer Modula 2

```

1  MODULE FourSquares;
2
3  (* ===== *)
4  (* Lagrange's Four Square Theorem - Standard Modula-2 *)
5  (* Every natural number can be expressed as the      *)
6  (* sum of at most four perfect squares.             *)
7  (* ===== *)
8
9  FROM InOut  IMPORT WriteString, WriteInt, WriteLn;
10 FROM MathLib0 IMPORT sqrt;
11
12 (* ----- *)
13 (* Procedure: FindFourSquares                        *)
14 (* Finds a, b, c, d >= 0 such that                  *)
15 (*   n = a*a + b*b + c*c + d*d                      *)
16 (* Sets found to TRUE if a decomposition is found. *)
17 (* ----- *)
18 PROCEDURE FindFourSquares (   n   : INTEGER;
19                             VAR a, b,
20                             c, d : INTEGER;
21                             VAR found : BOOLEAN);
22 VAR
23   sqrtN, remain, temp : INTEGER;
24   maxA, maxB, maxC    : INTEGER;
25   isqrt               : INTEGER;
26 BEGIN
27   found := FALSE;
28   a := 0;
29   b := 0;
30   c := 0;
31   d := 0;
32
33   (* Handle n = 0 *)
34   IF n = 0 THEN
35     found := TRUE;
36     RETURN
37   END;
38
39   (* Check if n is a perfect square (1 square) *)
40   sqrtN := TRUNC(sqrt(FLOAT(n)));
41   IF sqrtN * sqrtN = n THEN
42     a := sqrtN;
43     found := TRUE;
44     RETURN
45   END;
46
47   (* Check for sum of 2 squares *)
48   maxA := TRUNC(sqrt(FLOAT(n)));
49   a := 1;
50   WHILE a <= maxA DO
51     remain := n - a * a;

```

```

52     isqrt := TRUNC(sqrt(FLOAT(remain)));
53     IF isqrt * isqrt = remain THEN
54         b := isqrt;
55         found := TRUE;
56         RETURN
57     END;
58     INC(a)
59 END;
60
61 (* Check for sum of 3 squares. *)
62 (* Skip if n is of the form 4^k * (8m + 7), *)
63 (* which requires at least 4 squares. *)
64 temp := n;
65 WHILE temp MOD 4 = 0 DO
66     temp := temp DIV 4
67 END;
68
69 IF temp MOD 8 # 7 THEN
70     maxA := TRUNC(sqrt(FLOAT(n)));
71     a := 1;
72     WHILE a <= maxA DO
73         IF n - a * a <= 0 THEN
74             a := maxA + 1 (* break outer loop *)
75         ELSE
76             maxB := TRUNC(sqrt(FLOAT(n - a * a)));
77             b := a;
78             WHILE b <= maxB DO
79                 remain := n - a * a - b * b;
80                 IF remain < 0 THEN
81                     b := maxB + 1 (* break inner loop *)
82                 ELSE
83                     isqrt := TRUNC(sqrt(FLOAT(remain)));
84                     IF (isqrt * isqrt = remain) AND (isqrt >= b) THEN
85                         c := isqrt;
86                         found := TRUE;
87                         RETURN
88                     END;
89                     INC(b)
90                 END
91             END;
92             INC(a)
93         END
94     END
95 END;
96
97 (* Find sum of 4 squares (always possible by Lagrange's theorem) *)
98 maxA := TRUNC(sqrt(FLOAT(n)));
99 a := 1;
100 WHILE a <= maxA DO
101     IF n - a * a <= 0 THEN
102         a := maxA + 1 (* break outermost loop *)
103     ELSE
104         maxB := TRUNC(sqrt(FLOAT(n - a * a)));
105         b := a;

```

```

106     WHILE b <= maxB DO
107         IF n - a * a - b * b <= 0 THEN
108             b := maxB + 1 (* break loop *)
109         ELSE
110             maxC := TRUNC(sqrt(FLOAT(n - a * a - b * b)));
111             c := b;
112             WHILE c <= maxC DO
113                 remain := n - a * a - b * b - c * c;
114                 IF remain < 0 THEN
115                     c := maxC + 1 (* break innermost loop *)
116                 ELSE
117                     isqrt := TRUNC(sqrt(FLOAT(remain)));
118                     IF (isqrt * isqrt = remain) AND (isqrt >= c) THEN
119                         d := isqrt;
120                         found := TRUE;
121                         RETURN
122                     END;
123                     INC(c)
124                 END
125             END;
126             INC(b)
127         END
128     END;
129     INC(a)
130 END
131 END
132 END FindFourSquares;
133
134 (* ----- *)
135 (* Main module body *)
136 (* ----- *)
137 CONST
138     NTests = 6;
139
140 TYPE
141     TestArray = ARRAY [1..NTests] OF INTEGER;
142
143 VAR
144     testNumbers : TestArray;
145     n, a, b, c, d : INTEGER;
146     found        : BOOLEAN;
147     i            : INTEGER;
148
149 BEGIN
150     testNumbers[1] := 7;
151     testNumbers[2] := 15;
152     testNumbers[3] := 23;
153     testNumbers[4] := 100;
154     testNumbers[5] := 123;
155     testNumbers[6] := 999;
156
157     WriteString('=====');
158     WriteLn;
159     WriteString("Lagrange's Four Square Theorem - Modula-2");

```

```
160 WriteLn;
161 WriteString('=====');
162 WriteLn;
163 WriteLn;
164
165 FOR i := 1 TO NTests DO
166     n := testNumbers[i];
167     FindFourSquares(n, a, b, c, d, found);
168
169     IF found THEN
170         WriteString('n = ');
171         WriteInt(n, 0);
172         WriteString(' = ');
173         WriteInt(a, 0);
174         WriteString('^2 + ');
175         WriteInt(b, 0);
176         WriteString('^2 + ');
177         WriteInt(c, 0);
178         WriteString('^2 + ');
179         WriteInt(d, 0);
180         WriteString('^2 = ');
181         WriteInt(a*a + b*b + c*c + d*d, 0);
182         WriteLn
183     ELSE
184         WriteString('Could not find decomposition for ');
185         WriteInt(n, 0);
186         WriteLn
187     END
188 END
189
190 END FourSquares.
191
```