

Universidad Nacional Autónoma de México

Facultad de Ciencias

Física

Computación

Grupo 8108

Proyecto final

Integrantes:

Archundia Juárez Adrián

Duarte Martínez Jordán Aarón

Rodríguez Vega Nayeli

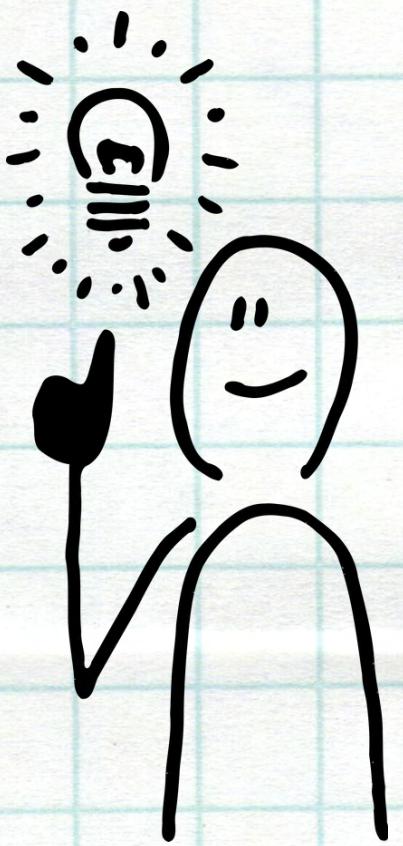
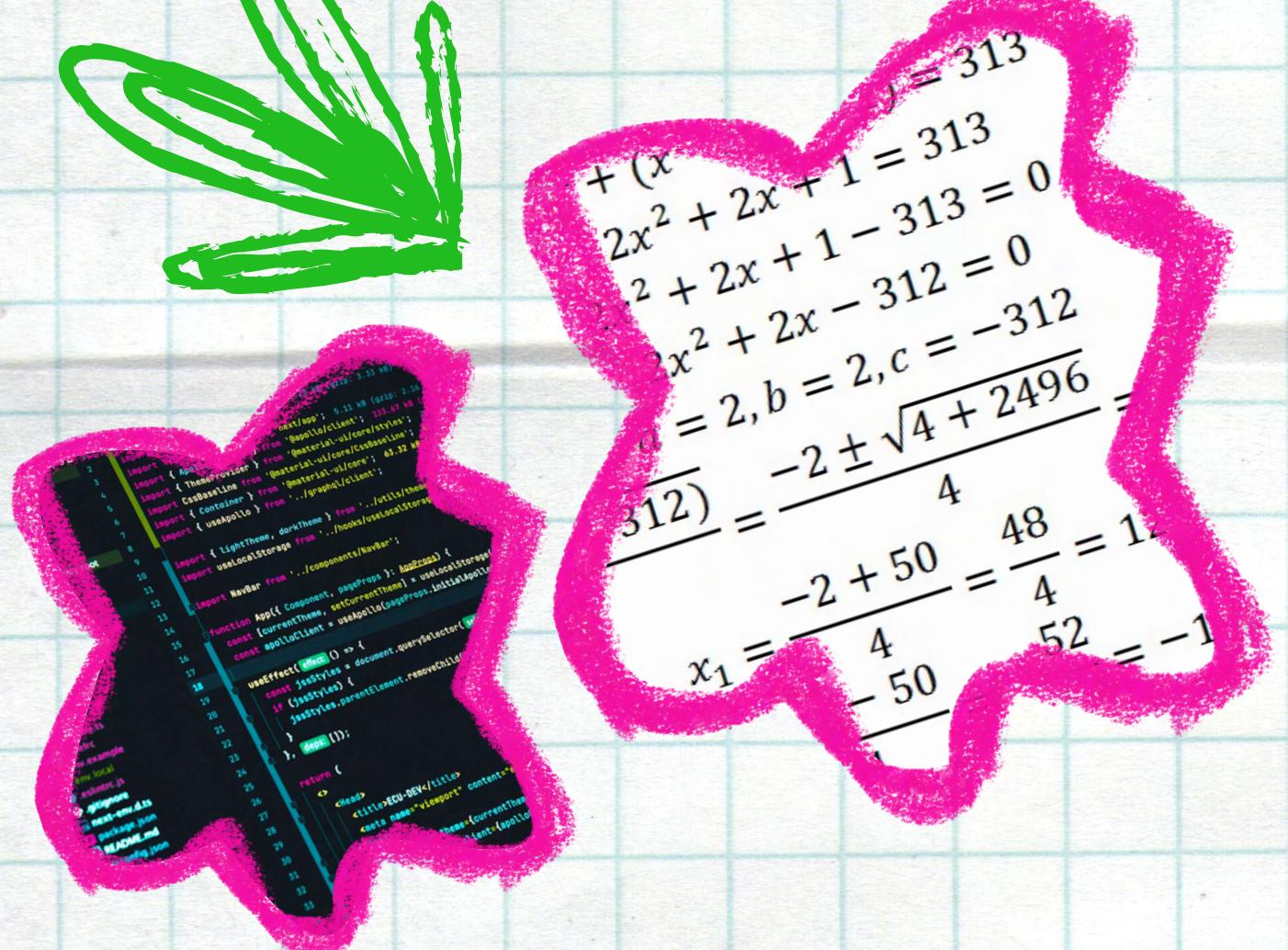


MACTZIL



CONTENIDO

- 1 Introducción
- 2 Objetivos
- 3 Desarrollo del experimento
- 4 Resultados
- 5 Conclusiones
- 6 Referencias



Introducción

Se allaron rasgos en donde los babilonios trataban de resolver ecuaciones



Ejipcios y el I método de la falsa posición

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Bhaskara, invención de la
formula general

$$x^3 + ax^2 + bx + c = 0$$

$$x \rightarrow y - \frac{a}{3}$$

$$y^3 + \frac{3b - a^2}{3}y + \frac{2a^3 - 9ab + 27c}{27} = 0$$

$y^3 + py = q$ (Cúbica reducida)

$$y = \sqrt[3]{\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

Método de Cardano

$$2x + 3y = 4$$

$$x^2 + 5x + 8 = 3$$

$$3x^3 + 3x^2 + 6x = 0$$

Algunos tipos de
ecuaciones

$$x \rightarrow y - \frac{a}{3}$$

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Objetivo

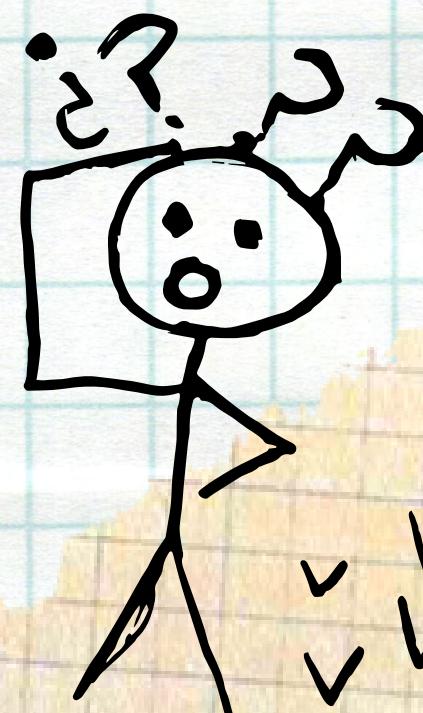
Resolver ecuaciones de segundo grado y de tercer grado con números complejos a través de nuestro programa llamado MACTZIL



$$z_1 = \left[\frac{-q}{2} + \sqrt{\frac{q^2}{4} + \frac{P^3}{27}} \right]^{\frac{1}{3}} + \left[\frac{-q}{2} - \sqrt{\frac{q^2}{4} + \frac{P^3}{27}} \right]^{\frac{1}{3}}$$

$$ax^3 + bx^2 + cx + d$$

$$\frac{y^3 + \frac{3b - a^2}{3}y + \frac{2a^3 - 9ab + 27c}{27}}{y^3 + py = q} = 0 \quad (\text{Cúbica reducida})$$



Preludio al PROGRAMA



Luca Pacioli

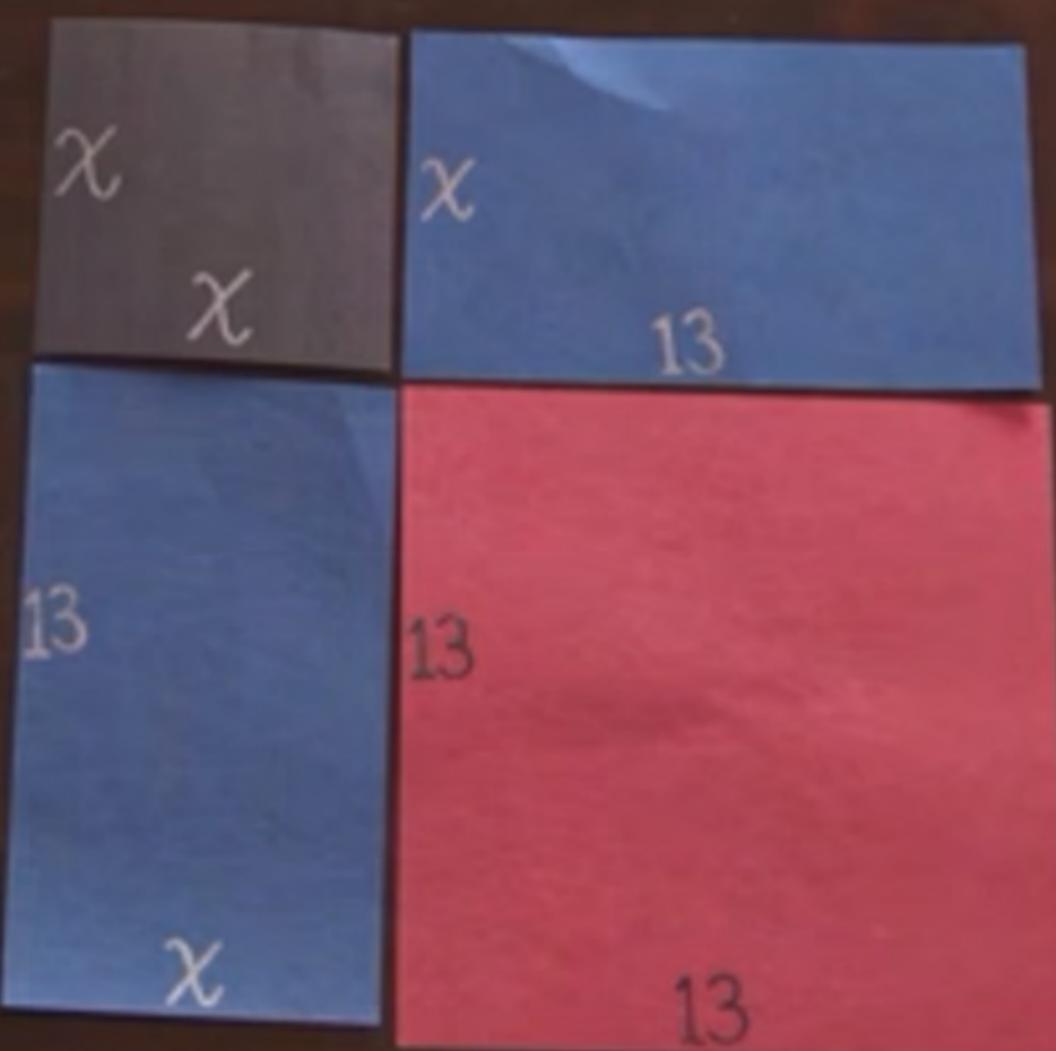
$$ax^3 + bx^2 + cx + d = 0$$



MAR

$$x^2 + 26x = 27$$

$$+ 169 \quad \quad \quad + 169$$



$$= 196$$

$$1. ax^2 = bx$$

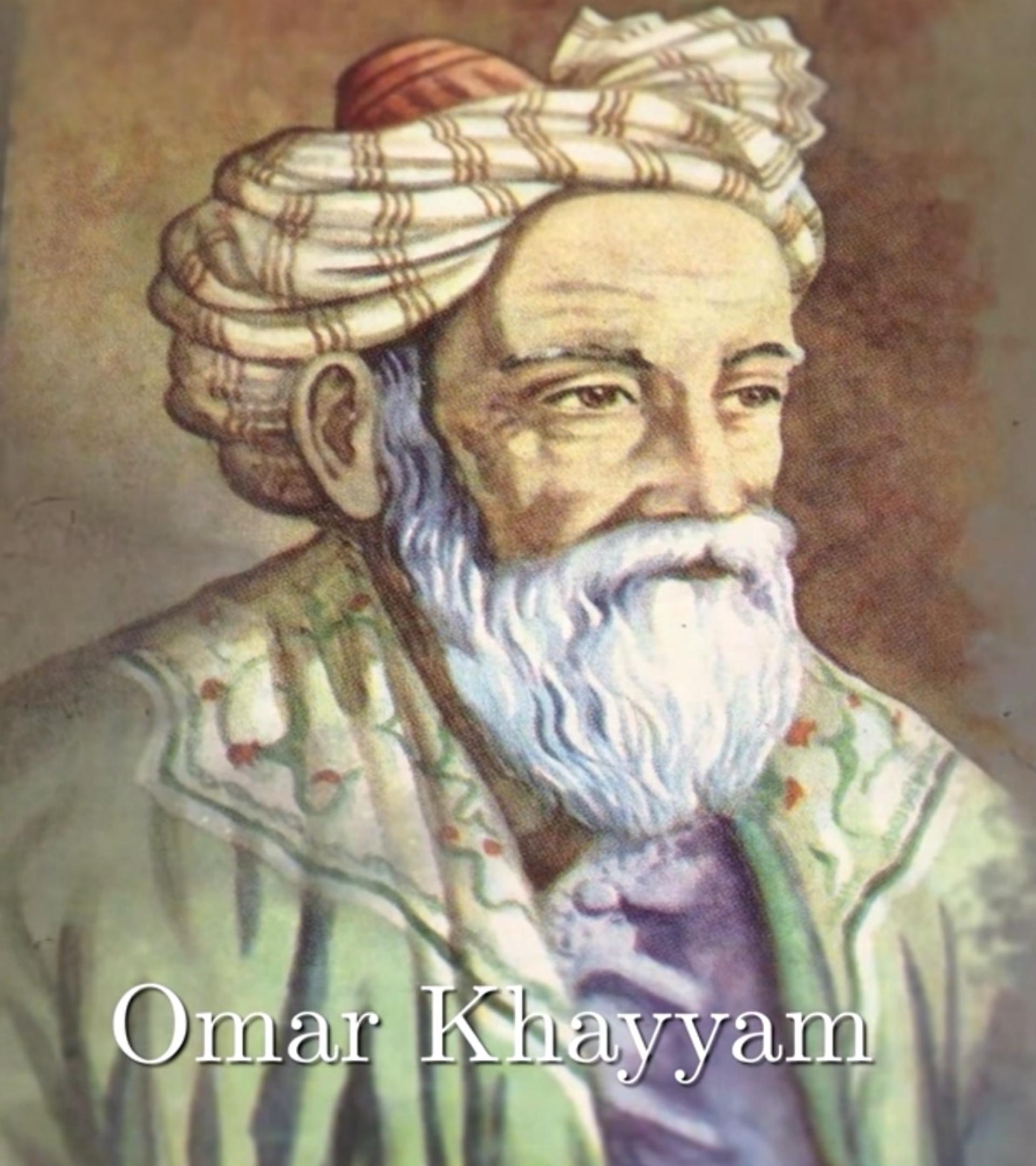
$$2. ax^2 = c$$

$$3. ax = c$$

$$4. ax^2 + bx = c$$

$$5. ax^2 + c = bx$$

$$6. bx + c = ax^2$$

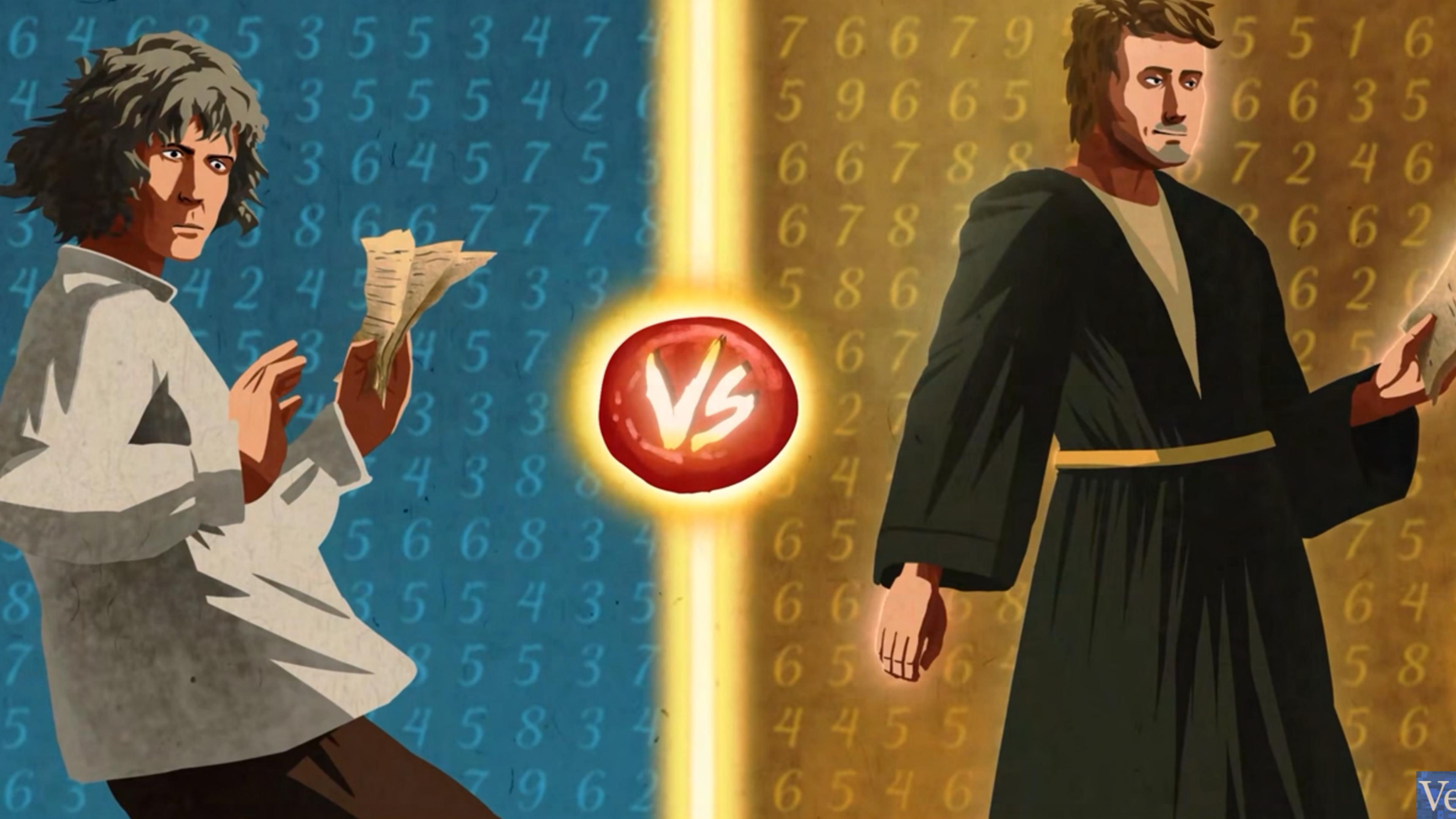


Omar Khayyam

$$\begin{aligned}x^3 &= c \\x^3 &= bx \\x^3 &= ax^2 \\x^3 + ax^2 &= bx \\x^3 &= ax^2 + bx \\x^3 + bx &= ax^2 \\x^3 + bx &= c \\x^3 + c &= bx \\x^3 + c &= ax^2 \\x^3 + ax^2 &= c \\x^3 &= ax^2 + c \\x^3 &= bx + c \\x^3 + ax^2 &= bx + c \\x^3 + c &= ax^2 + bx \\x^3 + c &= ax^2 + bx \\x^3 + ax^2 + c &= bx \\x^3 + ax^2 + bx &= c \\x^3 + bx &= ax^2 + c \\x^3 + bx + c &= ax^2\end{aligned}$$



Scipione Del Ferro

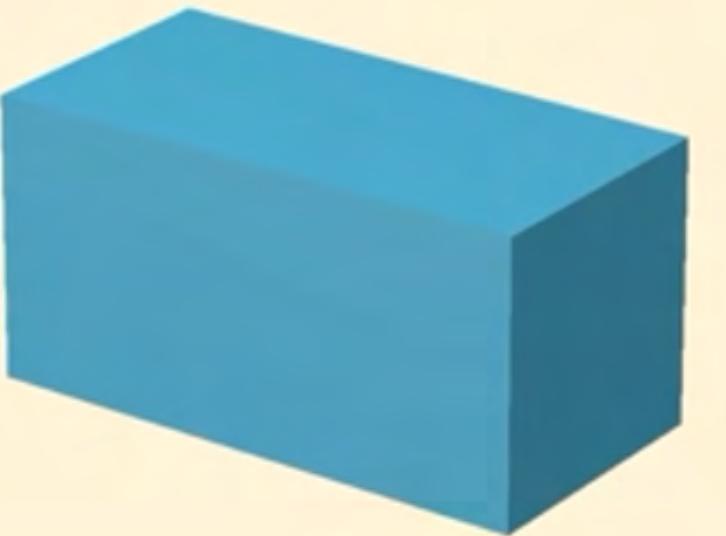
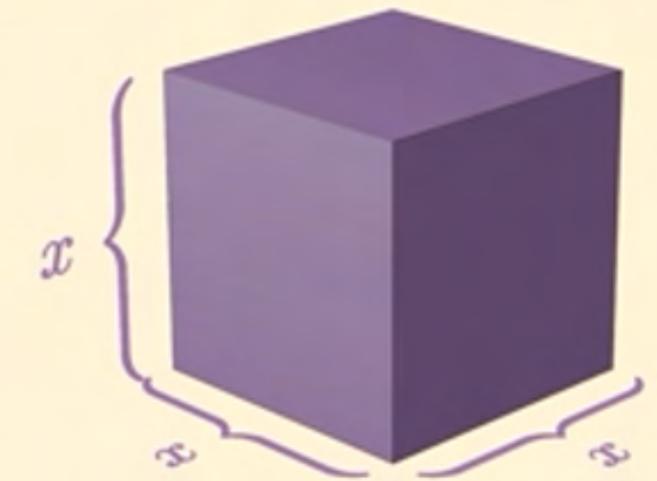


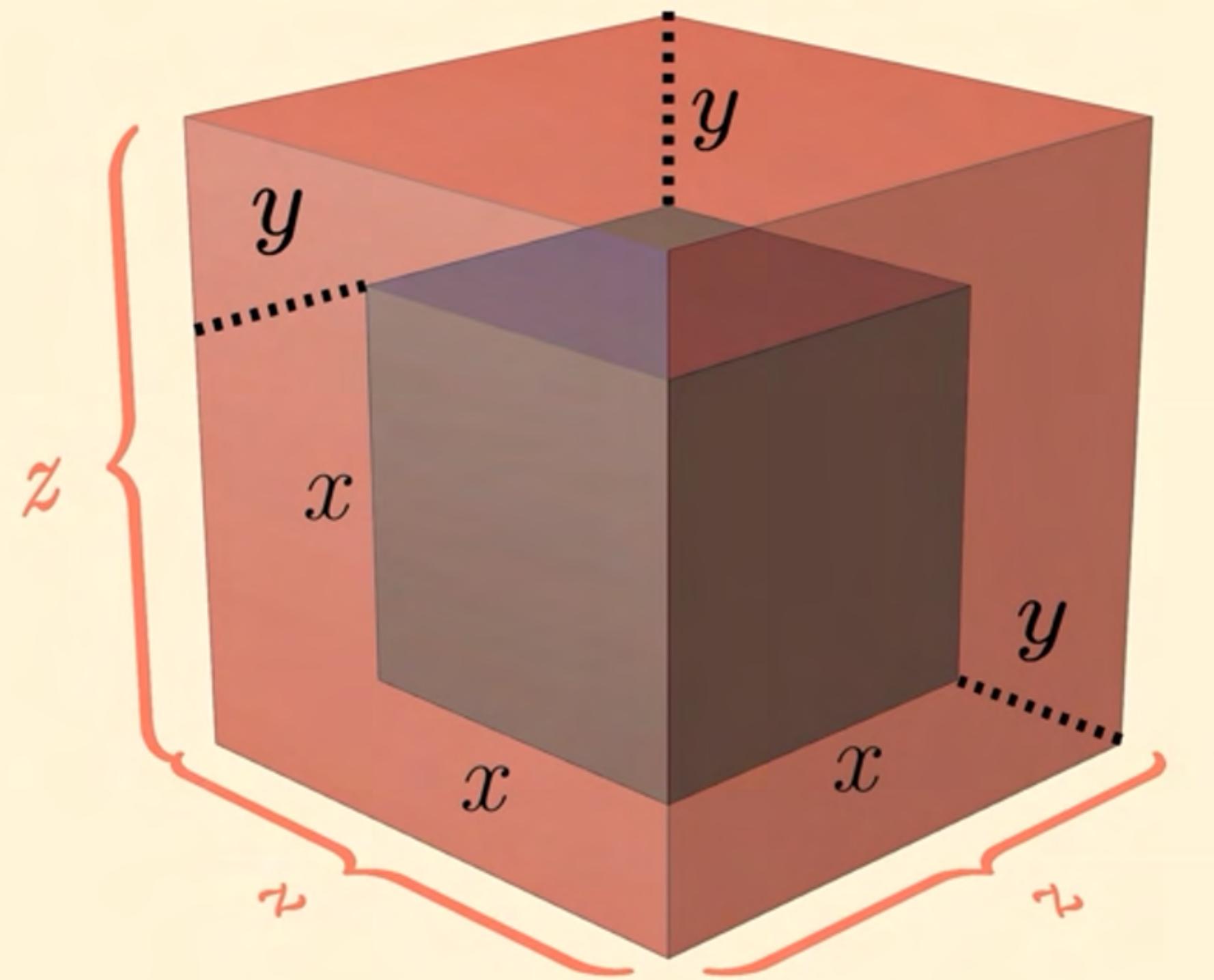
Ve



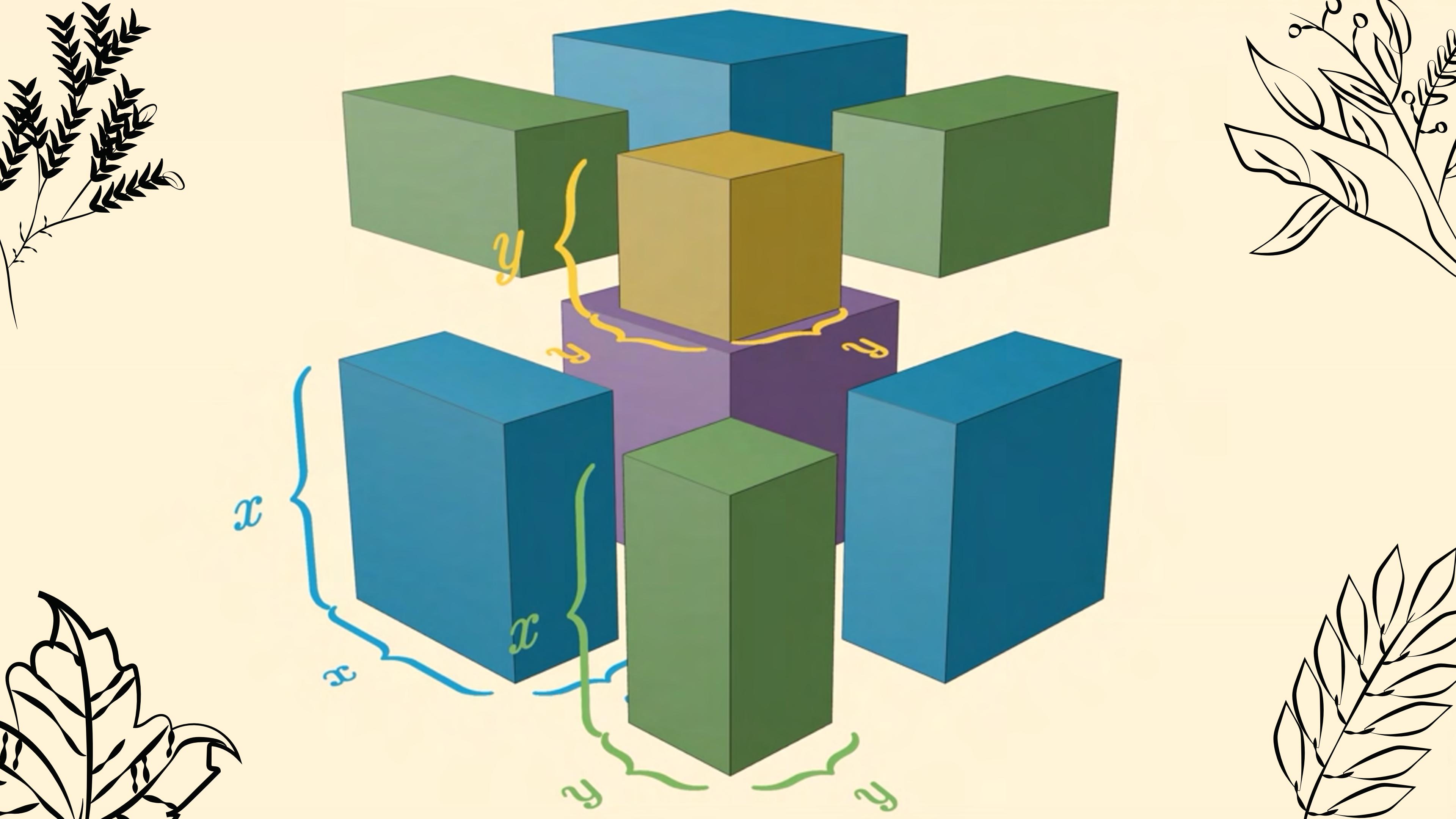
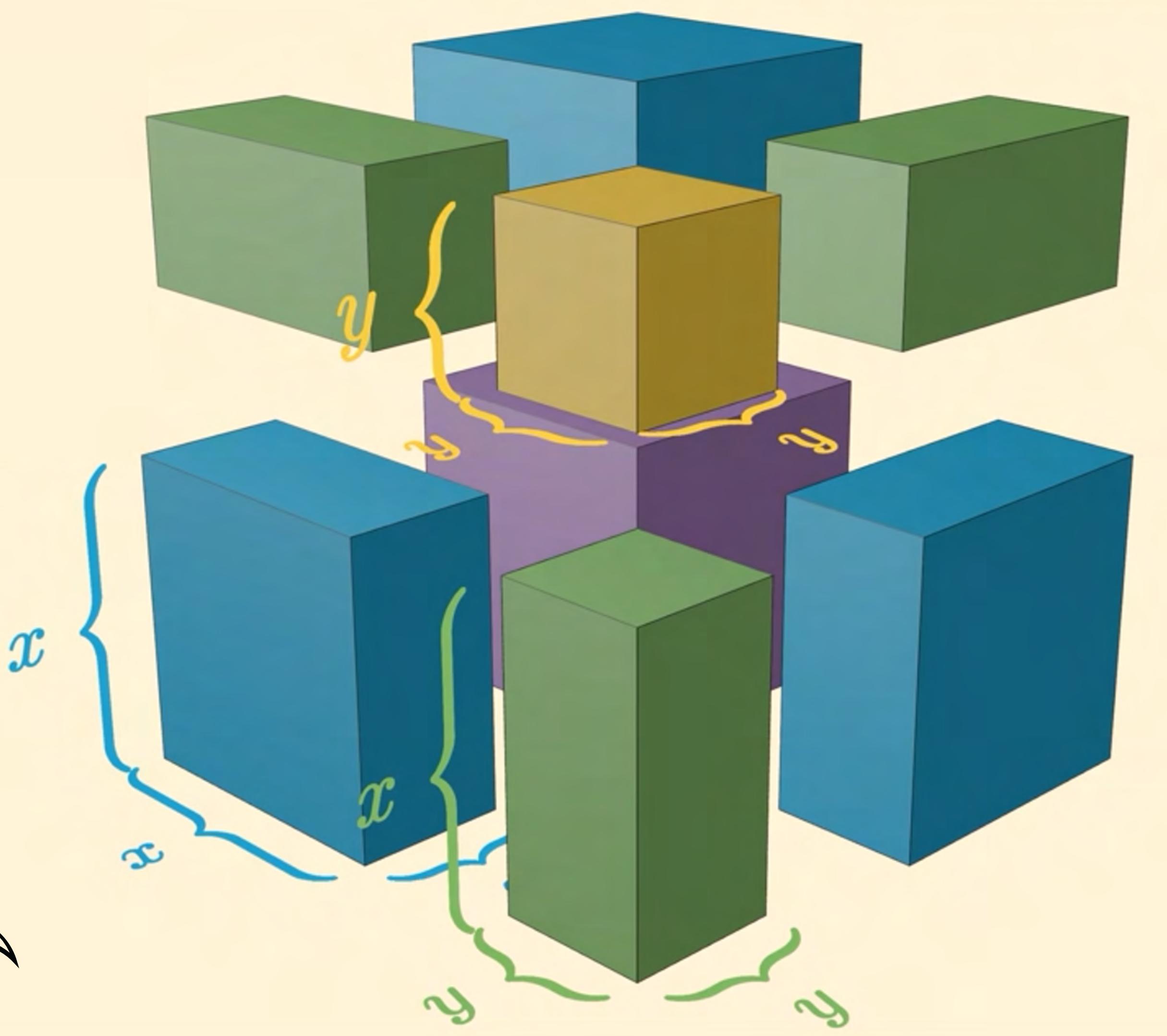
Niccolò Fontana Tartaglia

$$x^3 + 9x = 26$$



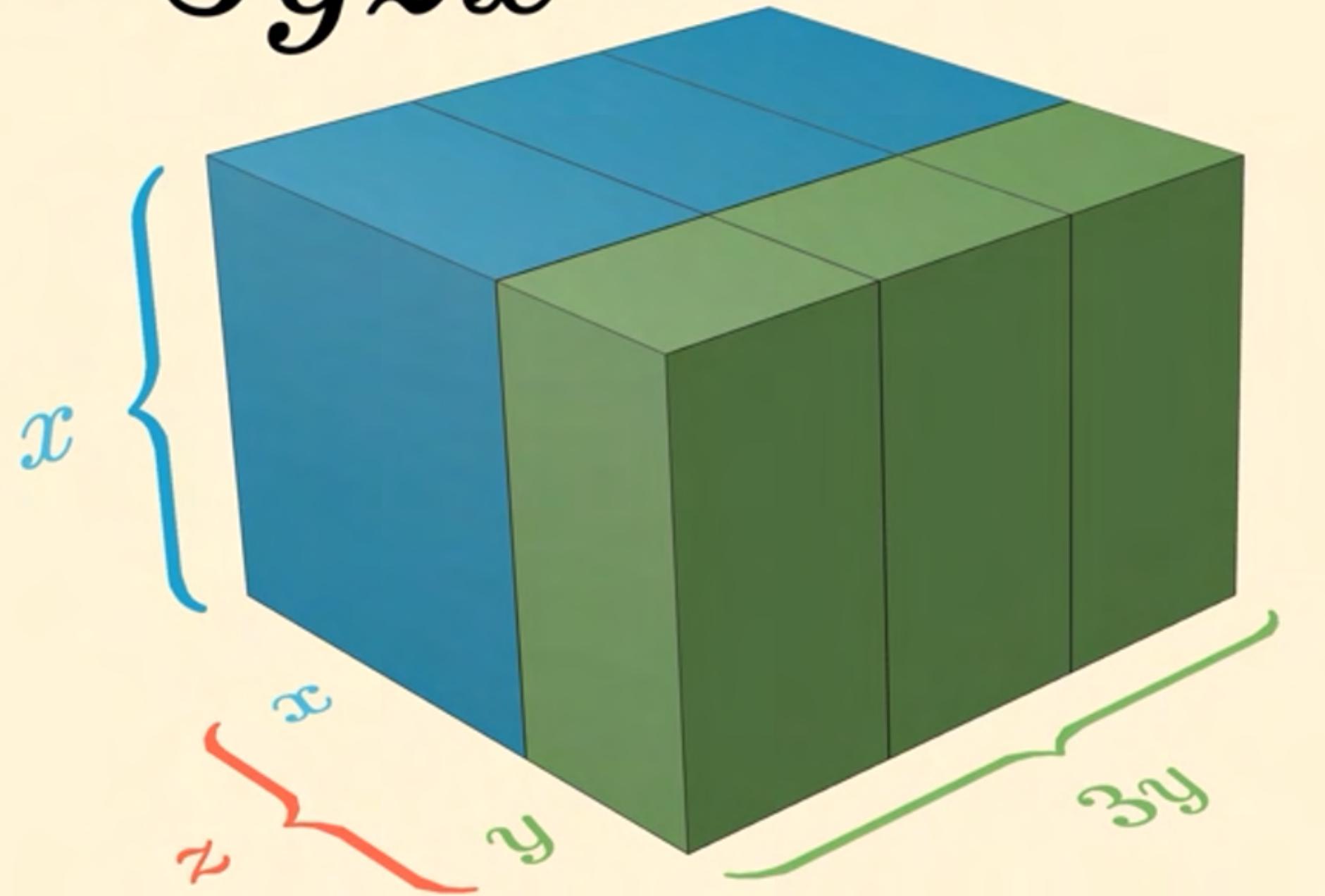


$$z = x + y$$



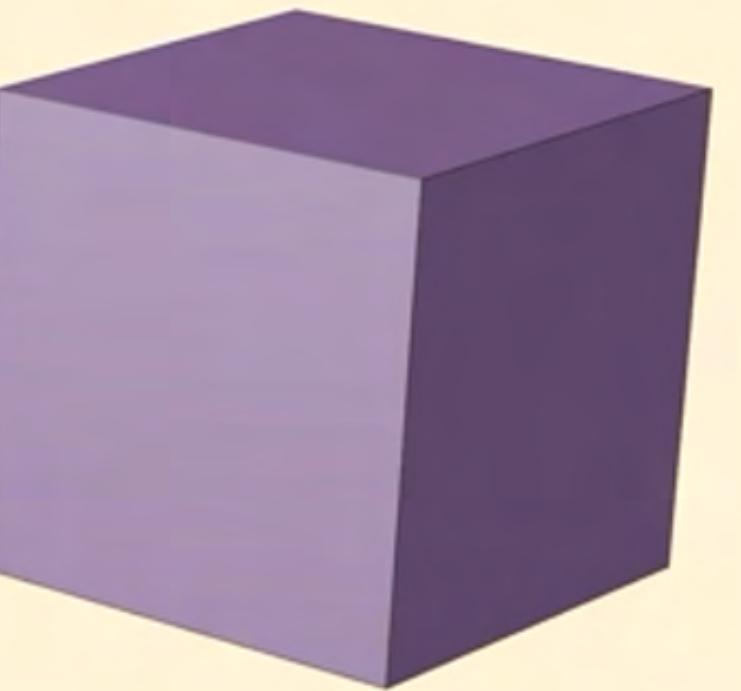
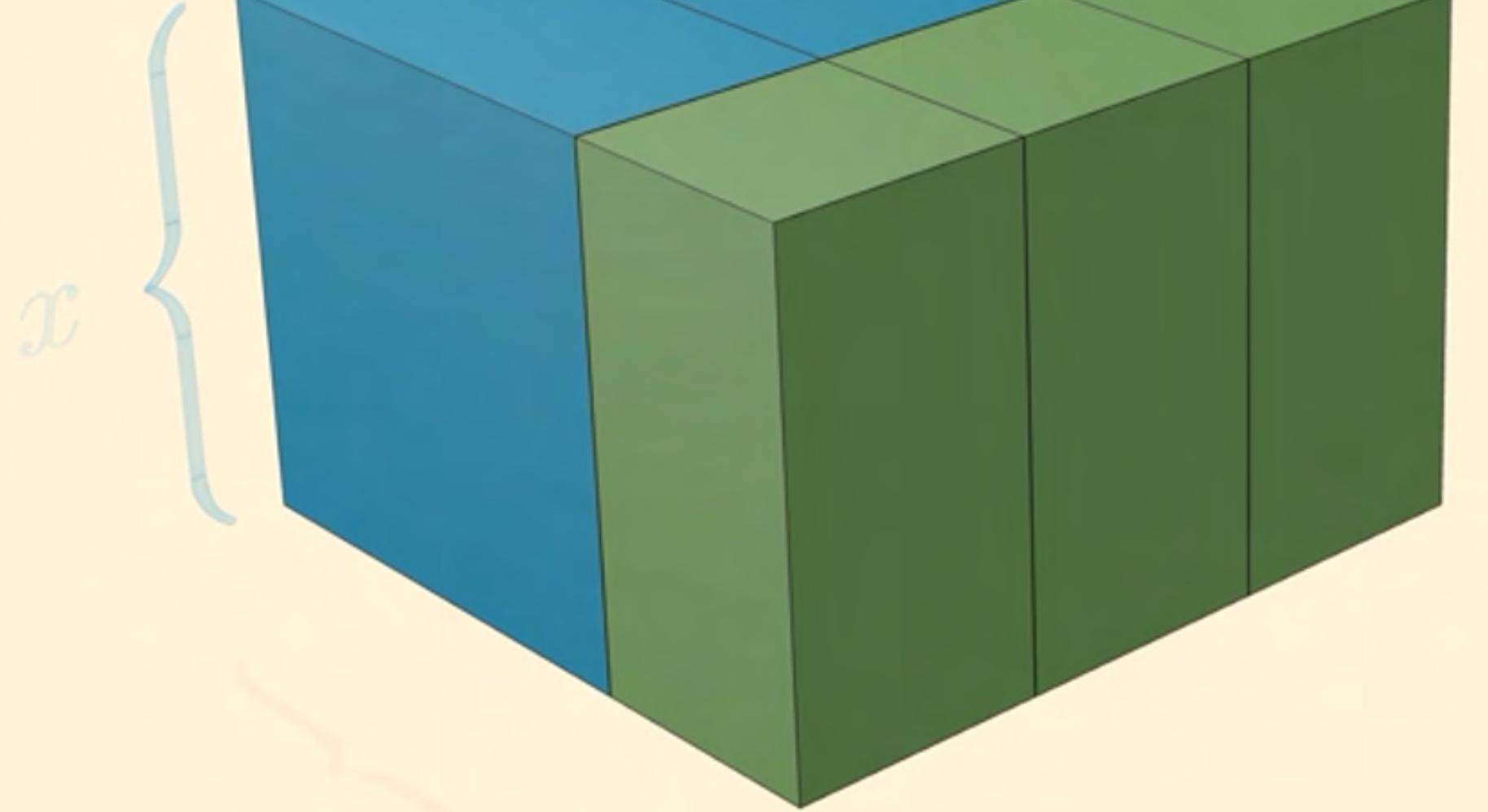
$$x^3 + 9x = 26$$

$$V = 3yzx$$

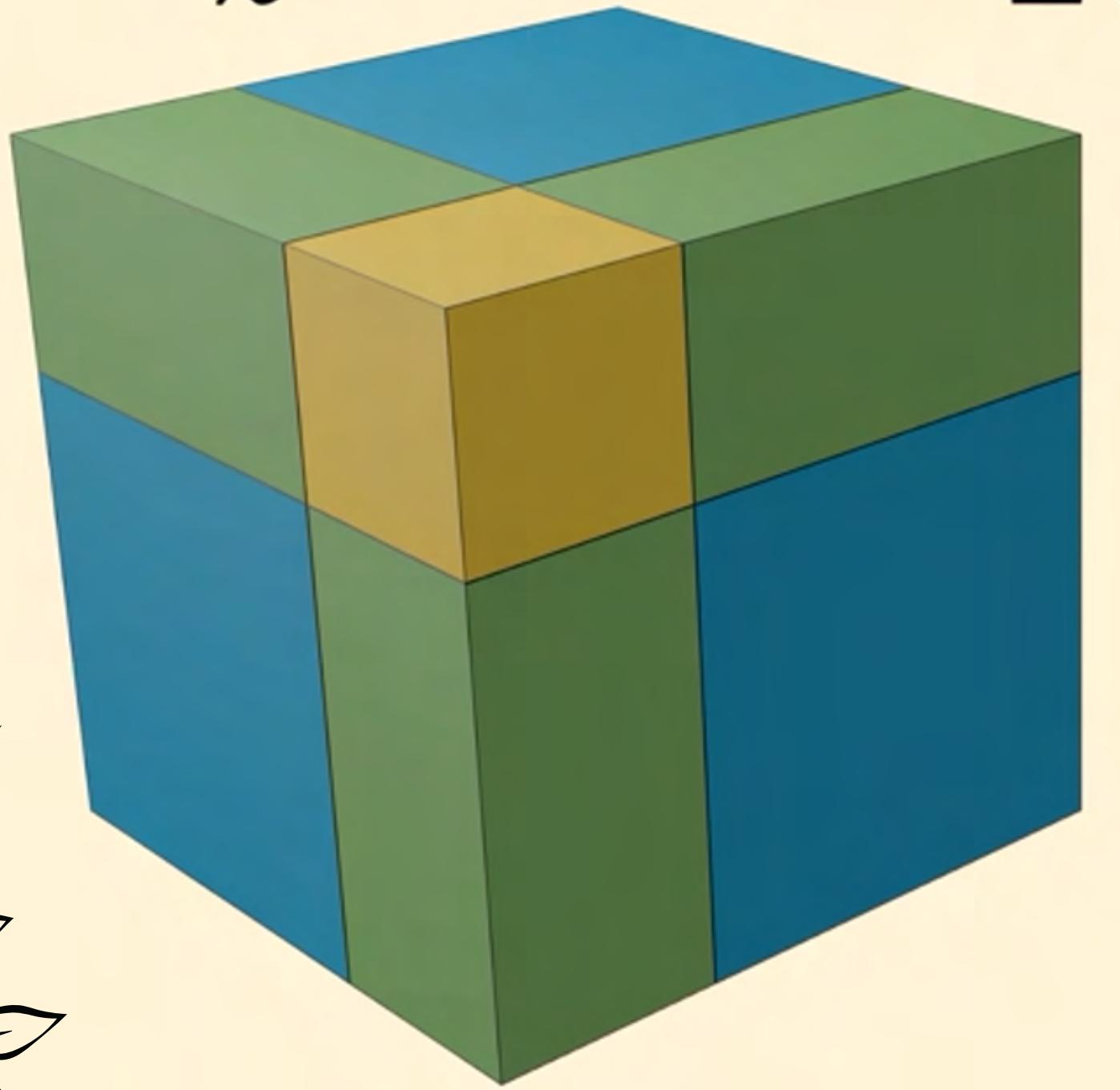


$$x^3 + 9x = 26$$

$$3yz = 9$$



$$\underbrace{x^3 + 9x + y^3}_{z^3} = 26 + y^3$$
$$= 26 + y^3$$



$$3yz = 9$$

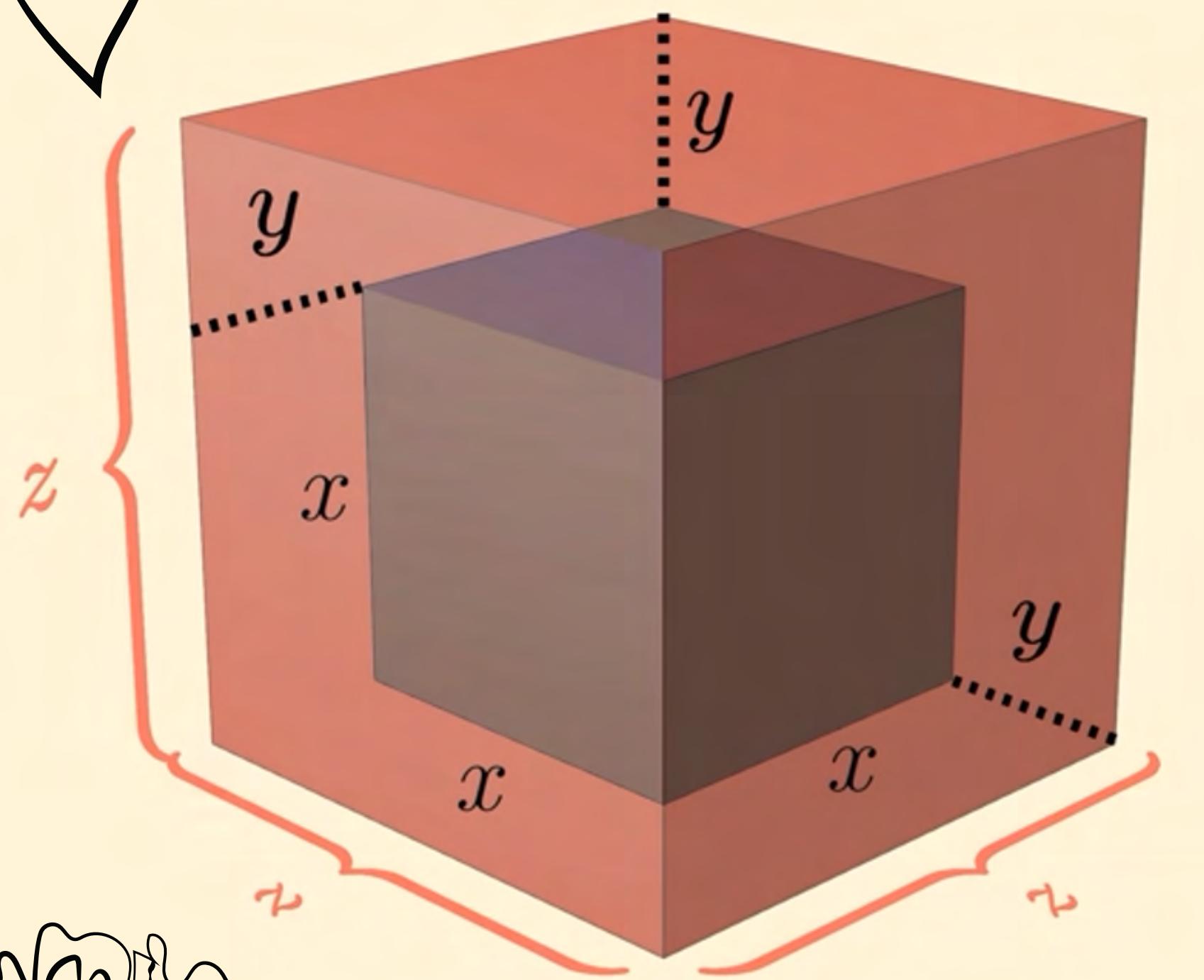
$$\underbrace{x^3 + 9x + y^3}_{z^3} = 26 + y^3$$

$$3yz = 9$$
$$z = \frac{3}{y}$$

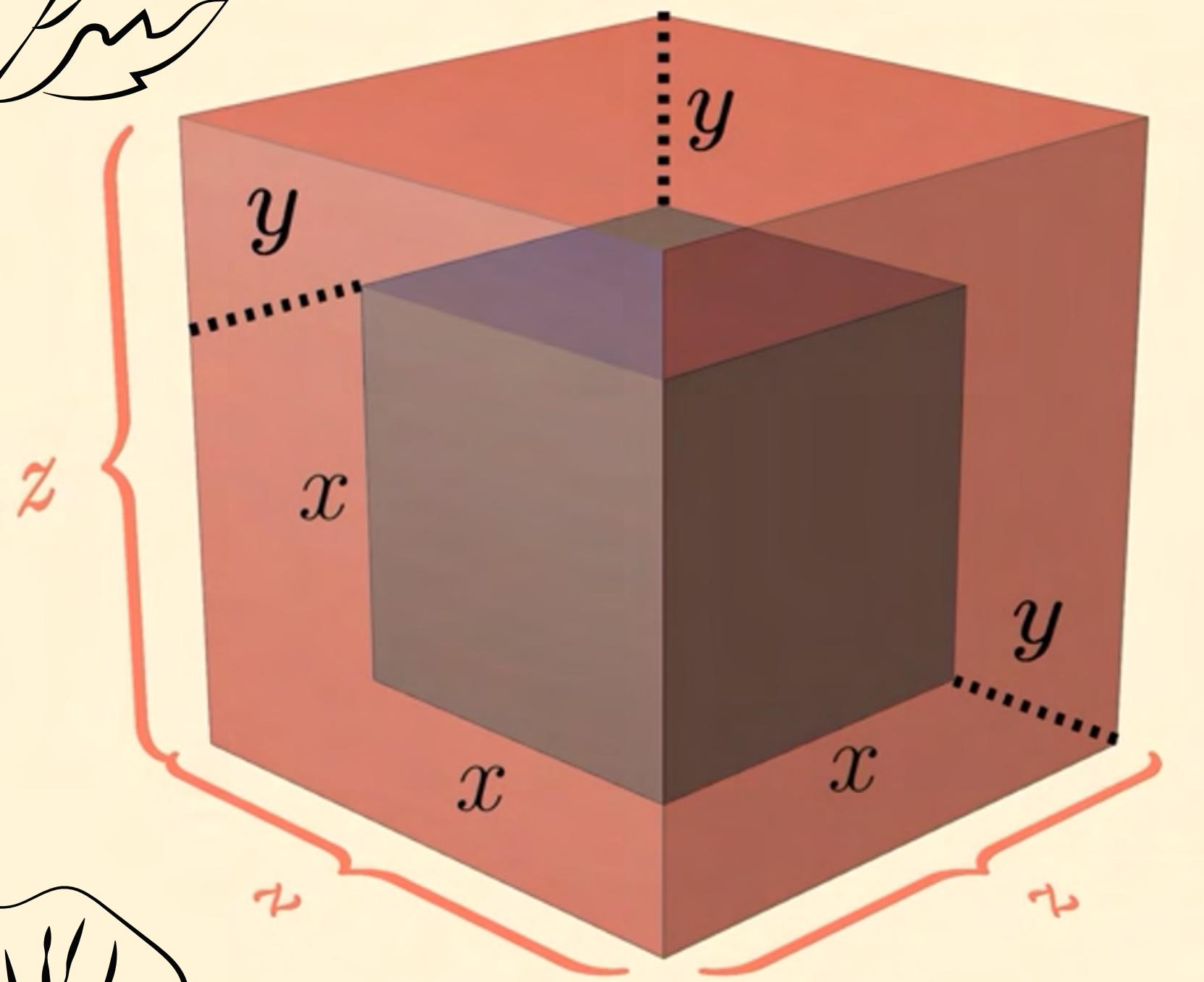
$$y^3 + 26 = \frac{27}{y^3}$$

$$y^6 + 26y^3 = 27$$

$$(y^3)^2 + 26(y^3) = 27$$



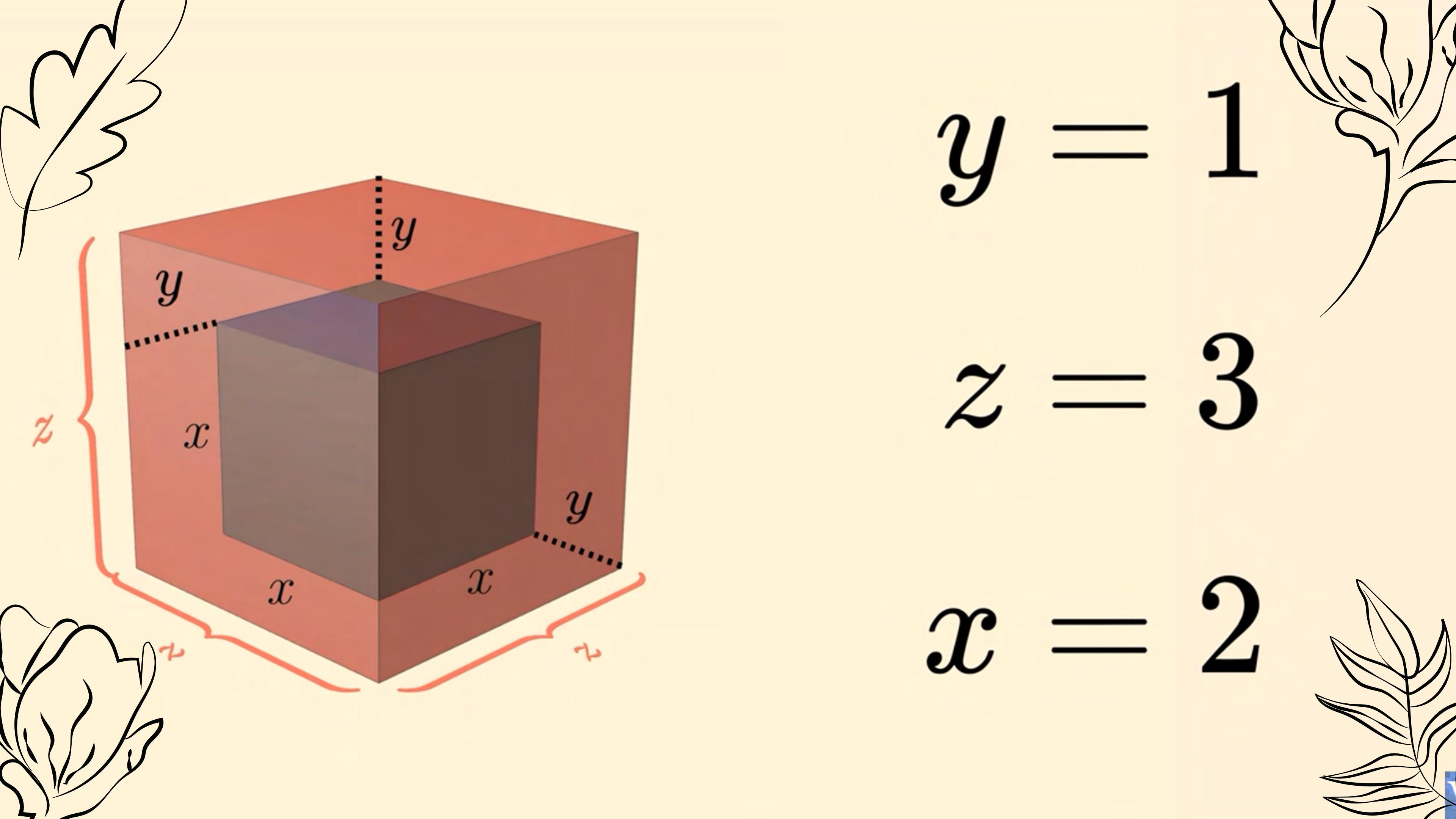
$$y = 1$$
$$z = -\frac{3}{y}$$



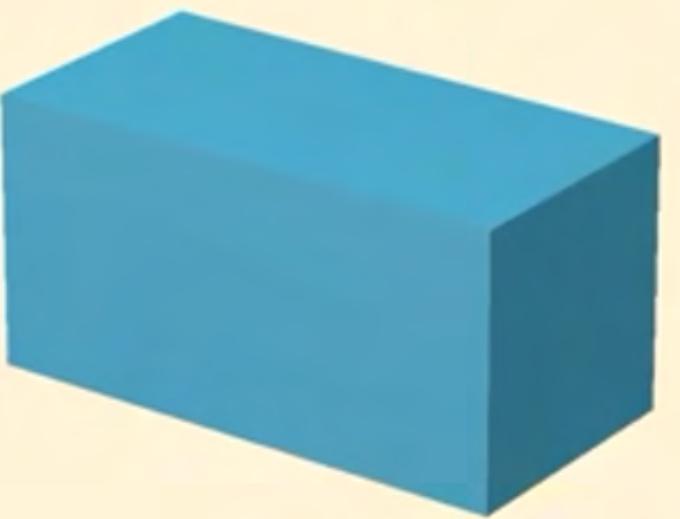
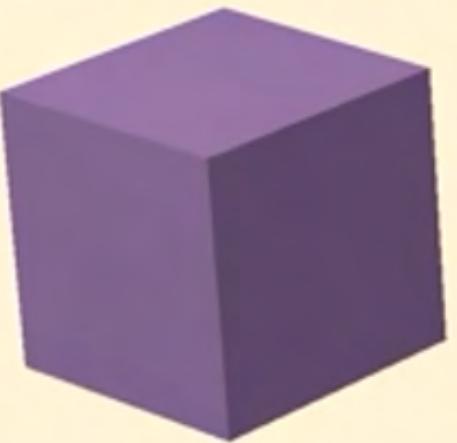
$$x + y = z$$

$$y = 1$$

$$z = 3$$



$$x^3 + 9x = 26$$



$$x^3 + cx = d : \sqrt[3]{\frac{d}{2} + \sqrt{\frac{d^2}{4} + \frac{c^3}{27}}} + \sqrt[3]{\frac{d}{2} - \sqrt{\frac{d^2}{4} + \frac{c^3}{27}}}$$

$$x^3 = cx + d : \sqrt[3]{\frac{d}{2} + \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}} + \sqrt[3]{\frac{d}{2} - \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}}$$

$$x^3 + d = cx : \sqrt[3]{-\frac{d}{2} + \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}} + \sqrt[3]{-\frac{d}{2} - \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}}$$

“Cuando el cubo con las cosas cerca,
se iguala a cualquier número discreto otros dos,
diferentes en eso encuentra.

Después por norma tendrás esto,
que su producto sea siempre igual al cubo
del tercio de las cosas neto;
después de su resultado general, de los lados del cubo,
bien restados verás tu cosa principal.

En el segundo, de los casos en acto; cuando el cubo se
queda solo, observarás este otro contrato del número
dos partes harás al vuelo tal que multiplicando la una
por la otra presto, el tercio del cubo de la cosa resulte
esto, de las cuales después, por común precepto, los
lados del cubo tomarás, juntos envueltos, y esa suma

será tu segundo concepto.
Los lados del cubo tomarás,
juntos envueltos,
y esa suma será tu segundo concepto.

El tercero, después de nuestras cuentas se resuelve
como el segundo, si bien vigila pues
son conjugados por naturaleza.

Esto encontré,
y no con paso lento en mil quinientos
cuatro y treinta,
con fundamentos firmes,
y sólidos,
en la ciudad que el mar rodea.



Gerolamo Cardano

$$a \left(x - \frac{b}{3a} \right)^3 + b \left(x - \frac{b}{3a} \right)^2 + c \left(x - \frac{b}{3a} \right) + d = 0$$

$$ax^3 \cancel{- \frac{2bx^2}{3}} + \frac{b^2x}{9} \cancel{- \frac{bx^2}{3}} + \frac{2b^2x}{3} - \frac{b^3}{27a} \cancel{+ bx^2} - \frac{2b^2x}{3a} + \frac{b^3}{9a^2} \cdots = 0$$

$$ax^3 + \left(c - \frac{b^2}{3a} \right) x + \left(d + \frac{2b^3}{27a^2} - \frac{bc}{3a} \right) = 0$$

$$x^3 = 15x + 4$$

$$\sqrt[3]{\frac{d}{2} + \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}} + \sqrt[3]{\frac{d}{2} - \sqrt{\frac{d^2}{4} - \frac{c^3}{27}}}$$

$$\sqrt[3]{\frac{4}{2} + \sqrt{\frac{4^2}{4} - \frac{15^3}{27}}} + \sqrt[3]{\frac{4}{2} - \sqrt{\frac{4^2}{4} - \frac{15^3}{27}}}$$

$$\sqrt[3]{2 + \sqrt{-121}} + \sqrt[3]{2 - \sqrt{-121}}$$

$$\sqrt[3]{2 + \sqrt{-121}} + \sqrt[3]{2 - \sqrt{-121}}$$

$$\sqrt[3]{2 + \sqrt{-121}} = a + b\sqrt{-1}$$

$$\sqrt[3]{2 - \sqrt{-121}} = a - b\sqrt{-1}$$

$$2 = a^3 - 3ab^2$$

$$11 = b(3a^2 - b^2)$$

$$\sqrt[3]{2 - \sqrt{-121}} = 2 - \sqrt{-1}$$

$$\sqrt[3]{2 + \sqrt{-121}} = 2 + \sqrt{-1}$$

$$2 + \sqrt{-1} + 2 - \sqrt{-1}$$
$$2 + 2 = 4$$

¿Y QUÉ PASA CON EL CODIGO?

**¿Qué no ese es el punto de
esta expo?**

¿Identificar grado?

Se dan las condiciones para identificar el grado del polinomio

```
1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
19
20        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
21        p = c/a
22        q = d/a
23
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se
```

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18
19    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
20
21        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
22        p = c/a
23        q = d/a
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```

¿Primer grado?

$$ax+b=0$$

Se usa la
fórmula:

$$x = -b/a$$

y se imprime
la raíz

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
19
20        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
21        p = c/a
22        q = d/a
23
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```

¿Segundo grado?

$$ax^2+bx+c=0$$

Se usa el
método de
Bhaskara:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
19
20        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
21        p = c/a
22        q = d/a
23
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```

¿Cúbica reducida?

$$ax^3+cx+d=0$$

Se divide todo entre " a " para adecuar los coeficientes al método de Cardano



```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18
19    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
20
21        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
22        p = c/a
23        q = d/a
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```

Pasito 2 (pa simplificar)

Se calcula
"U" y "V"

$$U = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{P^3}{27}}}$$

$$V = \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{P^3}{27}}}$$

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
19
20        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
21        p = c/a
22        q = d/a
23
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**1/2)/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**1/2)/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(z1) -(b/(3*a)) ) + ( (1j)*( ((3**1/2)/2)*(u-v) ) )
51        z3 = ( (-1/2)*(z1) -(b/(3*a)) ) - ( (1j)*( ((3**1/2)/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```



Passinho da morte

Se obtienen las raíces con:

$$Z_1 = U + V$$

$$Z_{2,3} = \left[-\frac{1}{2}(Z_1) \pm \sqrt{\frac{3}{4}(U-V)} \right] i$$



```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18
19    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
20
21        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
22        p = c/a
23        q = d/a
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(u+v) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
51        z3 = ( (-1/2)*(u+v) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```



¿Ec. cúbica?

$$ax^3+bx^2+cx+d=0$$

Se calculan los coeficientes para la forma reducida:

$$q = \frac{2b^3 - 9abc + 27a^2d}{27a^3}$$

$$p = \frac{3ac - b^2}{3a^2}$$

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18
19    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
20
21        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
22        p = c/a
23        q = d/a
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) )**(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(u+v) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
51        z3 = ( (-1/2)*(u+v) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```

Pasito 2.0 (pa simplificar)

Se calcula
"U" y "V" =

$$U = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{P^3}{27}}}$$

$$V = \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{P^3}{27}}}$$

```

1 def chicharronera_madre(a,b,c,d):
2
3     if a==0 and b==0: #Si la ec. es de primer grado, se usa fórmula para la ec. de primer grado
4
5         x = -d/c
6         print("La raíz es: {}".format(x)) #Se imprime "La raíz es x".
7
8
9
10    elif a==0: #Si la ec. es cuadrática, se procede a usar la fórmula de Bhaskara.
11
12        x1 = ( -c + ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad positiva de la fórmula se le asigna el valor x1.
13        x2 = ( -c - ( (c**2 -(4*b*d))**(1/2) ) ) / (2*b) #A la posibilidad negativa de la fórmula se le asigna el valor x2.
14
15        print("Las raíces son: {} y {}".format(x1,x2)) #Se imprime "Las raíces son x1, x2".
16
17
18
19    elif b==0: #Si la ec. es una cúbica reducida, se procede a usar la fórmula de Cardano.
20
21        #Se calculan los coeficientes "p" y "q" cuando se obliga el valor de a=1.
22        p = c/a
23        q = d/a
24
25        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
26        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
27        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
28
29        #Se calculan todas las raíces según sus fórmula dadas por Cardano.
30        z1 = u+v
31        z2 = ( (-1/2)*(z1) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
32        z3 = ( (-1/2)*(z1) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
33
34        #Se imprimen las raíces junto con un mensaje de advertencia.
35        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
36        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"
37
38    else: #Si la ec. es una cúbica, se procede a usar la fórmula de cardano mediante la reducción de a cúbica reducida.
39
40        #Se calculan los coeficientes "p" y "q" para la forma reducida.
41        p = ( (3*a*c) - (b**2) )/( 3*(a**2) )
42        q = ( (3*(b**3)) - (9*a*b*c) + (27*(a**2)*d) )/( (27*(a**3)) )
43
44        #Se dan a los valores "u" y "v" correspondientes según la fórmula de Cardano.
45        u = ( -(q/2) + ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
46        v = ( -(q/2) - ( ( (q**2)/4 ) + ((p**3)/27) )**(1/2) ) **(1/3)
47
48        #Se calculan todas las raíces según sus fórmula dadas por Cardano (con su modificación para ser más exacto).
49        z1 = u+v -(b/(3*a))
50        z2 = ( (-1/2)*(u+v) -(b/(3*a)) ) + ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
51        z3 = ( (-1/2)*(u+v) -(b/(3*a)) ) - ( (1j)*( ((3**((1/2))/2)*(u-v) ) )
52
53        #Se imprimen las raíces junto con un mensaje de advertencia.
54        print ("Las raíces son: {}, {} y {}".format(z1,z2,z3))
55        print ("Debido al uso de varias raíces para la obtención de la fórmula, es posible que los valores proporcionados se"

```



Passinho da morte 2.0

Se calculan las raíces con:

$$X_1 = U + V - \frac{b}{3a}$$

$$X_{2,3} = \left[-\frac{1}{2}(U+V) - \frac{b}{3a} \right]$$

$$\pm \left[\frac{\sqrt{3}}{2}(U-V) \right] i$$

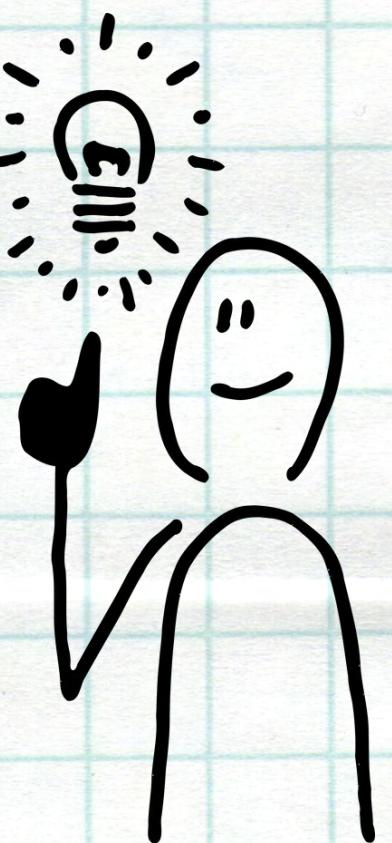


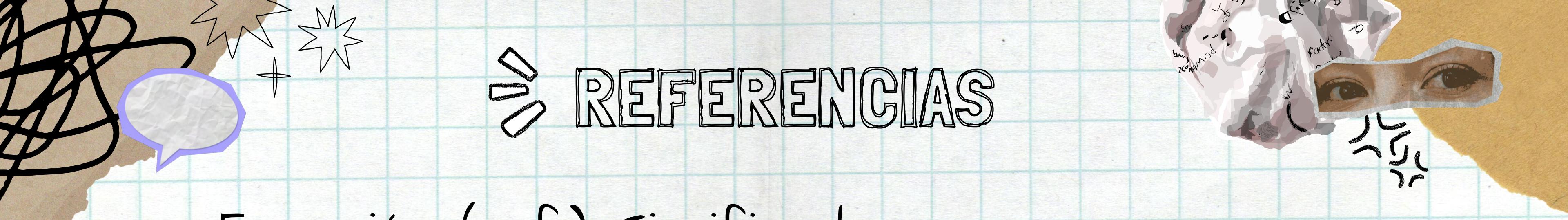
Resultados

Logramos obtener un código nos permite conocer de forma inmediata los resultados de ecuaciones cuadráticas o cubicas, asimismo, si el resultado arroja una solución compleja, lo resuelve por igual. También es importante recalcar que el código es capaz de resolver la ecuación sin que esta completa, también selecciona automáticamente el método o la vía por donde va a resolver la ecuación, dependiendo del tipo de ecuación en cuestión. Sin embargo, es importante mencionar que posiblemente por defectos de aproximación del sistema operativo, o quizás por la naturaleza de la formula usada para la resolución de ecuaciones cúbicas, hay ocasiones en las que la calculadora no arroja resultados correctos, siendo en su mayoría aproximaciones buenas, y en ocasiones completos errores

CONCLUSIONES

El código puede ser de gran ayuda, ya que, hace la función de una calculadora, en donde obtienes resultados de polinomios no triviales muy fácilmente, y este puede ser empleado en distintos campos donde se haga uso de el, haciendo que el trabajo de dichas ecuaciones sea mas sencillo y que de esta forma, ayude a la eficacia de trabajos y/o proyectos.





» REFERENCIAS

- Ecuación. (s. f.). Significados. <https://www.significados.com/ecuacion/>
- Sobre el origen de las ecuaciones. (s. f.). Junta de extremadura.
- Veritasium en español. (12 de diciembre de 2021). Cómo se Inventaron los Números Imaginarios [Video]. YouTube. <https://www.youtube.com/watch?v=VN7nipynEOc>

GRACIAS POR SU ATENCIÓN !

¿Alguna pregunta?