1

a, b = map(int, input().split())

if a != 0:

   x = - b / a

elif b == 0:

   print("VSN")

else:

   print("VN")

2

a, b = map(int, input().split())

if a > 0:

   print(f"x > - {b} / {a}")

elif a < 0:

   print(f"x < - {b} / {a}")

else:

   if b > 0:

       print("VN")

   else:

       print("VSN")

3

import math

a, b, c = map(int, input().split())

delta = b \* b - 4 \* a \* c

if delta > 0:

   x1 = (-b + math.sqrt(delta))/(2 \* a)

   x2 = (-b - math.sqrt(delta))/(2 \* a)

   print(f"x1 = {x1}, x2 = {x2}")

elif delta == 0:

   x = - b / (2 \* a)

   print(f"x = {x}")

else:

   print("VN")

4

def giai\_phuong\_trinh\_bac\_hai\_no\_condition(a, b, c):

    if a == 0:

        if b != 0:

            return f'Phương trình trở thành phương trình bậc nhất: x = {-c / b}'

        elif c == 0:

            return 'Phương trình có vô số nghiệm'

        else:

            return 'Phương trình vô nghiệm'

    else:

        return giai\_phuong\_trinh\_bac\_hai(a, b, c)

# Ví dụ

a = 0

b = 2

c = -4

print(giai\_phuong\_trinh\_bac\_hai\_no\_condition(a, b, c))

5

import numpy as np

def giai\_he\_phuong\_trinh(A, B):

    try:

        # A là ma trận hệ số, B là ma trận hằng số

        X = np.linalg.solve(A, B)

        return f'Nghiệm của hệ: x = {X[0]}, y = {X[1]}'

    except np.linalg.LinAlgError:

        return 'Hệ phương trình vô nghiệm hoặc có vô số nghiệm'

# Ví dụ

A = np.array([[1, 2], [3, 4]])

B = np.array([5, 6])

print(giai\_he\_phuong\_trinh(A, B))

6

def tinh\_tong\_1(n):

    S = 0

    for i in range(1, n + 1):

        S += i \* 3

    return S

# Ví dụ

n = 5

print(tinh\_tong\_1(n))

7

def tinh\_tong\_2(n):

    S = 0

    for i in range(1, n + 1):

        S += 1 / (i \* (i + 1))

    return S

# Ví dụ

n = 5

print(tinh\_tong\_2(n))

8

def tinh\_tong\_3(x, n):

    S = 0

    for i in range(1, n + 1):

        S += x \*\* i

    return S

# Ví dụ

x = 2

n = 5

print(tinh\_tong\_3(x, n))

9

def tinh\_tong\_4(x, n):

    S = 1  # bắt đầu với 1 (tương đương với x^0)

    for i in range(1, n + 1):

        S += x \*\* i

    return S

# Ví dụ

x = 2

n = 5

print(tinh\_tong\_4(x, n))

11

import math

def tinh\_tong\_factorial\_1(n):

    S = 0

    for i in range(1, n + 1):

        S += math.factorial(i)

    return S

# Ví dụ

n = 5

print(f"Tổng S = 1 + 2! + 3! + ... + {n}! là:", tinh\_tong\_factorial\_1(n))

12

def tinh\_tong\_factorial\_2(n):

    S = 1  # bắt đầu với 1 (tương đương với 0!)

    for i in range(1, n + 1):

        S += math.factorial(2 \* i + 1)

    return S

# Ví dụ

n = 3

print(f"Tổng S = 1 + 3! + 5! + ... + (2\*{n}+1)! là:", tinh\_tong\_factorial\_2(n))

16

a, b, c = map(int, input().split())

if a+b>c and a+c>b and b+c>a:

   print("là tam giác")

else:

   print("không phải tam giác")

17

x1, y1 = map(float, input().split())

x2, y2 = map(float, input().split())

x3, y3 = map(float, input().split())

area = 0.5 \* abs(x1 \* (y2 - y3) + x2 \* (y3 - y1) + x3 \* (y1 - y2))

if area == 0:

   print("Ba điểm thẳng hàng, không tạo thành tam giác.")

else:

   print("Ba điểm tạo thành một tam giác với diện tích:", area)