from threading import Thread

import utils

def ndPolynomialEval(inString):

threads = []

ndSoln = utils.NonDetSolution()

inputs = inString.split(";")

params1 = inputs[0]

params1 = params1.split(' ')

params2 = inputs[1]

params2 = params2.split(' ')

length = len(params2)

for i in range(length):

t = Thread(target = polynomialEval, args = (int(params2[i]), params1, ndSoln))

threads.append(t)

solution = utils.waitForOnePosOrAllNeg(threads, ndSoln)

return solution

# return output

def polynomialEval(param, coefficients, ndSoln):

sum = 0

length = len(coefficients)

for i in range(length):

sum += int(coefficients[i]) \* ( param \*\* (length - i - 1))

if(sum == 0):

ndSoln.setSolution(str(param))

print(ndPolynomialEval("1 0 -7 6;4 -3 2 5"))

print(ndPolynomialEval("1 0 -7 6;4 5"))

print(ndPolynomialEval("1 1 1 1 1 1;1 -1"))