import math

from pdb import pm

from sympy import li

from sympy.ntheory.factor\_ import totient

from sympy import primefactors, factorint

class EulerPhiFunctions:

    def \_\_init\_\_(self, n = 1,a = 1):

        self.n = n

        self.a = a

    def isPrimeNumber(self,num):

        if(num < 2):

            return False

        squareRoot = int(math.sqrt(num))

        for i in range(2, squareRoot +1 ):

            if(num % i == 0):

                return False

        return True

    def listPrimeNumber(self):

        listn =""

        if(self.n >= 2):

            listn += "2" + " "

        for i in range(3, self.n+1):

            if(self.isPrimeNumber(i)):

                listn += str(i) + " "

            i+=2

        print(listn)

    def EulerPhi(self,n):

        return totient(n)

    def checkPrimeTGT(self):

        if(math.gcd(self.n,self.a)) == 1:

            print("nguyen to cung nhau")

        else:

            print("ko nguyen to cung nhau")

    def AmoduleN(self):

        return self.a\*\*(self.EulerPhi(self.n)-1)

    def PTPrime(self,n):

        i = 2

        listN = []

        while(n > 1):

            if(n % i == 0):

                n = int(n/i)

                listN.append(i)

            else:

                i+=1

        if(len(listN)== 0):

            listN.append(n)

        return listN

    def Print\_PTPrime(self):

        lN = self.PTPrime(self.n)

        size = len(lN)

        sb = ""

        for i in range(0, size - 1):

            sb += str(lN[i]) + " \* "

        sb += str(lN[size - 1])

        print("Result:",self.n,"=",sb)

    def Print\_Prime\_Euler(self):

        arr\_prime = primefactors(self.n)

        print("ϕ(%s) = %s " % (self.n, self.n),end='')

        for i in range(0, len(arr\_prime)):

            print("\* (1 - 1/%s)" % (arr\_prime[i]),end='')

n = int(input("Nhap so nguyen duong: "))

a = int(input("Nhap so nguyen to cung nhau: "))

Pm = EulerPhiFunctions(n,a)

Pm.listPrimeNumber()

print(Pm.EulerPhi(n))

# Pm.checkPrimeTGT()

print(Pm.AmoduleN())

Pm.Print\_PTPrime()

Pm.Print\_Prime\_Euler()

