#Bài 1

import math

class MyPoint2D:

  def \_\_init\_\_(point, x, y):

    point.x = x

    point.y = y

  def distance(point, p: 'MyPoint2D'):

        return math.sqrt((point.x - p.x) \*\* 2 + (point.y - p.y) \*\* 2)

  def \_\_repr\_\_(point):

        return f"x={point.x}, y={point.y}"

x1 = int(input("Nhập x1:"))

y1 = int(input("Nhập y1:"))

p1 = MyPoint2D(x1, y1)

print(f"Point 1: {p1}")

x2 = int(input("Nhập x2:"))

y2 = int(input("Nhập y2:"))

p2 = MyPoint2D(x2, y2)

print(f"Point 2: {p2}")

dist = p1.distance(p2)

print(f"Khoảng cách: {dist}")

#Bài 2

class MyTime:

  def \_\_init\_\_(time, hour:int, minute:int, second:int):

    time.hour = hour

    time.minute = minute

    time.second = second

  def display(time):

       if (hour <= 12):

            return str(f"Time: {time.hour:02}:{time.minute:02}:{time.second:02} AM")

       elif (hour > 12):

            return str(f"Time: {time.hour:02}:{time.minute:02}:{time.second:02} PM")

  def add\_hours(time, hours:int):

        time.hour = (time.hour + hours) % 24

  def add\_minute(time, minutes:int):

        total\_minutes = time.minute + minutes

        time.minute = total\_minutes % 60

        time.hour = (time.hour + total\_minutes // 60) % 24

  def add\_seconds(time, seconds:int):

        total\_seconds = time.second + seconds

        time.second = total\_seconds % 60

        total\_minutes = time.minute + total\_seconds // 60

        time.minute = total\_minutes % 60

        time.hour = (time.hour + total\_minutes // 60) % 24

  def \_\_repr\_\_(time):

        if (hour <= 12):

            return f"Time: {time.hour}:{time.minute}:{time.second} AM"

        elif (hour > 12):

            return f"Time: {time.hour}:{time.minute}:{time.second} PM"

hour = int(input("Nhập giờ:"))

minute = int(input("Nhập phút:"))

second = int(input("Nhập giây:"))

t = MyTime(hour,minute,second)

print(f"{t.display()}")

h = int(input("Nhập giờ cần thêm:"))

t.add\_hours(h)

print(f"Thời gian sau khi thêm {h} giờ: {t.display()}")

m = int(input("Nhập phút cần thêm:"))

t.add\_minute(m)

print(f"Thời gian sau khi thêm {m} phút: {t.display()}")

s = int(input("Nhập giây cần thêm:"))

t.add\_seconds(s)

print(f"Thời gian sau khi thêm {s} giây: {t.display()}")

#Bài 3

class MyDate:

    def \_\_init\_\_(date, day:int, month:int, year:int):

        date.day = day

        date.month = month

        date.year = year

    def display(date):

        return f"{date.day:02}/{date.month:02}/{date.year:04}"

    def \_\_repr\_\_(date):

        return f"Date: {date.day:02}/{date.month:02}/{date.year:04}"

    def add\_years(date, years:int):

        date.year += years

    def add\_months(date, months:int):

        date.year += (date.month + months - 1) // 12

        date.month = (date.month + months - 1) % 12 + 1

    def add\_days(date, days:int):

        while days > 0:

            days\_in\_current\_month = date.days\_in\_month(date.month, date.year)

            if date.day + days <= days\_in\_current\_month:

                date.day += days

                days = 0

            else:

                days -= (days\_in\_current\_month - date.day + 1)

                date.day = 1

                date.add\_months(1)

    def days\_in\_month(date, month:int, year:int):

        if month in {1, 3, 5, 7, 8, 10, 12}:

            return 31

        elif month in {4, 6, 9, 11}:

            return 30

        else:

            if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

                return 29

            else:

                return 28

day = int(input("Nhập ngày:"))

month = int(input("Nhập tháng:"))

year = int(input("Nhập năm:"))

when = MyDate(day, month, year)

print(f"{when.display()}")

y = int(input("Nhập số năm cần thêm:"))

when.add\_years(y)

print(f"Thời điểm sau khi thêm {y} năm: {when.display()}")

m = int(input("Nhập số tháng cần thêm:"))

when.add\_months(m)

print(f"Thời điểm sau khi thêm {m} tháng: {when.display()}")

d = int(input("Nhập số ngày cần thêm:"))

when.add\_days(d)

print(f"Thời điểm sau khi thêm {d} ngày: {when.display()}")

#Bài 7

import json

class InputData:

    def \_\_init\_\_(self, file\_name):

        self.file\_name = file\_name

    def read\_elements(self):

        with open(self.file\_name, 'r') as file:

            num\_elements = int(file.readline().strip())

            elements = [file.readline().strip() for \_ in range(num\_elements)]

        return elements

    def convert\_to\_dictionary(self, elements):

        return {f"element {i+1}": element for i, element in enumerate(elements)}

    def convert\_to\_json(self, dictionary):

        return json.dumps(dictionary)

    def write\_to\_file(self, data, output\_file):

        with open(output\_file, 'w') as file:

            file.write(data)

class InputSpecialData(InputData):

    def \_\_init\_\_(self, file\_name):

        super().\_\_init\_\_(file\_name)

    def thong\_ke(self, dictionary):

        even\_elements = {key: value for key, value in dictionary.items() if isinstance(value, int) and value % 2 == 0}

        return even\_elements

input\_data = InputSpecialData('BT.txt')

elements = input\_data.read\_elements()

dictionary = input\_data.convert\_to\_dictionary(elements)

json\_data = input\_data.convert\_to\_json(dictionary)

input\_data.write\_to\_file(json\_data, 'output.json')

even\_elements = input\_data.thong\_ke(dictionary)

print("Các phần tử có giá trị là số chẵn:")

for key, value in even\_elements.items():

    print(f"{key}: {value}")