import json

import datetime

OPEN\_TIME = 10

CLOSE\_TIME = 21

def check\_date(str\_date):

if len(str\_date.split("-")) != 3:

return False

year, month, day = str\_date.split("-")

if len(year) != 4 or len(month) != 2 or len(day) != 2:

return False

if not(year.isdigit() and month.isdigit() and day.isdigit()):

return False

year, month, day = int(year), int(month), int(day)

if 2000 <= year <= 3000 and 1 <= month <= 12 and 1 <= day <= 31:

return True

else:

return False

def ask\_date():

print("Введите дату:")

date = None

while date is temp:

else:

date = None

temp = input()

if check\_date(temp):

date = temp

else:

print("Введена не коректная дата")

return date

def list\_stat(A):

stat = {}

for time in range(OPEN\_TIME, CLOSE\_TIME + 1):

stat[time] =0

for number in A:

if number in stat:

stat[number] += 1

else:

stat[number] = 1

return stat

def list\_stat\_2(A):

stat = {}

for number in set(A):

if number in stat:

stat[number] = A.count(number)

return stat

def list\_sort(A):

stat = list\_stat(A)

keys = list(stat.keys())

keys.sort(key = lambda x: stat[x])

print(keys)

def load\_info():

with open("info.json") as f:

info = json.load(f)

return info

info = load\_info()

print(info)

def compare\_dates(date\_1, date\_2, date\_3):

date\_1 = datetime.date.fromisoformat(date\_1)

date\_2 = datetime.date.fromisoformat(date\_2)

date\_3 = datetime.date.fromisoformat(date\_3)

return date\_1 <= date\_2 <= date\_3

def calc\_mean\_wait(stat):

s = 0

cnt = 0

for record in stat:

s += record["wait"]

cnt += 1

mean = s / cnt

print(f"Среднее время ожидания: {mean}")

def calc\_optimization(stat):

counter = {}

for time in range(OPEN\_TIME, CLOSE\_TIME + 1):

counter[time] = 0

for record in stat:

counter[record["hour"]] += 1

hours = list(counter.items())

hours.sort(key=lambda x: x[1])

print("Часы работы от наименьшей загруженности до наибольшей:")

for hour, workload in hours:

print(f"{hour}:00 - {workload} человек")