# def sum\_list(numbers):

#     return sum(numbers)

# print (sum\_list([1,2,3,4,5 ]))

# for i in range(1,11):

#     print(i)

# import csv

# with open("large\_file.csv", "r") as csvfile:

#     reader = csv.DictReader(csvfile)

#     for row in reader:

#         print(f"Name: {row['Name']}, Age: {row['Age']}")

#         if reader.line\_num > 5:

#             break

# import re

# with open("emails.txt", "r") as file:

#     text = file.read()

# emails = re.findall(r'[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}', text)

# print("Found emails:", emails)

# import psutil

# p = psutil.Process()

# with p.oneshot():

#     p.name()  # collect multiple info

#     print(p.cpu\_times())  # return cached value

#     print()

# import os

# with open('/tmp/timestamp.log') as f:

#     lines = f.readlines

# with open('/tmp/warning.txt', 'w') as f:

#      for line in lines:

#        if  ' warning' in line:

#         f.write(line)

# ===================================================

# import configparser

# config = configparser.ConfigParser()

# config.read('/tmp/config.ini')

# host = config['DATABASE'] ['host']

# user = config['DATABSASE'] ['user']

# print("Host:" , host)

# print("user:" , user)

# ==================================================

# question 3

# import os

# file\_path = '/tmp/report.csv'

# if os.path.exists(file\_path):

#     with open(file\_path) as f:

#         for i in range(5):

#             line = f.readline()

#             if not line:

#                 break

#             print(line.strip())

# else:

#     print("file not found: /tmp/report.csv")

# #================================================================

# question 4

# import psutil

# cpu\_usage = psutil.cpu\_percent(interval=1)

# print(f"cpu usage:{cpu\_usage}%")

# if cpu\_usage > 80:

#     print ("high cpu utilization detected")

# ================================================

# question 5

# import subprocess

# with open('/tmp/servers.txt') as f:

#     servers = f.read().splitlines()

# for server in servers:

#      result = subprocess.run(['ping', '-c', '1', server],

#                              stdout = subprocess.DEVNULL,

#                              stderr=subprocess.DEVNULL,)

#      if result.returncode == 0:

#          print(f"{server}: ping successful")

#      else:

#          print(f"{server}: ping failed")

#===========================================================

# question 6

# unique\_timestamps = set()

# with open('/tmp/access.log') as f :

#     for line in f:

#         timestamp = line.split([0])

#         unique\_timestamps.add(timestamp)

# timestamps\_list = list(unique\_timestamps)

# print(timestamps\_list)

#==================================================

# question 7

# import json

# with open('tmp/user\_data.json') as f:

#     data = json.load(f)

# for user in data:

#     if user['age'] > 28:

#         print (user['name'])

# =======================================

# question 8

# import shutil

# total, used, free = shutil.disk\_usge('/tmp')

# free\_gb = free / (1024 \*\* 3)

# print(f"free space: {free\_gb:.2f} GB")

# if free\_gb < 1:

#     print("low disk space on /tmp!")

# ==============================================

# question 9

# values = {

#     "name" : "user123",

#     "status": "active",

#     "link": "hhtp://example.com/login",

#     }

# with open('/tmp/email\_templete.txt') as f:

#     template = f.read()

# for key, val in values.items():

#     templete = templete.replace(f"{{{{key}}}}", val)

# print(templete)

# ===========================================================

#question 10

# count = 0

# with open('/tmp/security.log') as f:

#     for line in f:

#         count+= line.lower().count('failed')

# print(f"number of failed occurence: {count}")