Exercise 1:

import numpy as np

import cv2

my\_img = np.zeros((400, 400, 3), dtype = "uint8")

cv2.rectangle(my\_img, (30, 30), (300, 200), (0, 20, 200), 10)

cv2.circle(my\_img, (200, 200), 80, (0, 20, 200), 10)

cv2.line(my\_img, (202, 220), (100, 160), (0, 20, 200), 10)

cv2.imshow('Window', my\_img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Exercise 2:

# a – Telnet to the server

import getpass

import sys

import telnetlib

host = "192.168.1.101"

port=22

timeout =10

user = raw\_input("Enter your remote account: ")

password = getpass.getpass()

tn = telnetlib.Telnet(host)

tn.read\_until("login: ")

tn.write(user + "\n")

if password:

tn.read\_until("Password: ")

tn.write(password + "\n")

tn.write("ls\n")

tn.write("exit\n")

print tn.read\_all()

# b – SSH to server

from paramiko import SSHClient

ssh = SSHClient()

ssh.load\_system\_host\_keys()

ssh.connect('127.0.0.1',5000)

ssh\_stdin, ssh\_stdout, ssh\_stderr = ssh.exec\_command('ls')

print(ssh\_stdout)

# f – Copy files to remote server

from paramiko import SSHClient

from scp import SCPClient

ssh = SSHClient()

ssh.load\_system\_host\_keys()

ssh.connect('127.0.0.1')

with SCPClient(ssh.get\_transport()) as scp:

scp.put('my\_file.txt', 'my\_file.txt')

Exercise 3:

App.py :

import sqlite3

from flask import jsonify, Flask,request

from celery import Celery

from flask\_celery import make\_celery

import json

import requests,os

app = Flask(\_\_name\_\_)

@app.route("/sql\_connection/" , methods=["POST"])

def sql\_connection():

request\_data = request.json

print(request\_data)

con = sqlite3.connect('mydatabase.db')

cursorObj = con.cursor()

cursorObj.execute(

"CREATE TABLE mytable

(id SERIAL NOT NULL PRIMARY KEY

, ipaddress varchar”

))

cursorObj.execute(

"INSERT INTO mytable(id, ipaddress) VALUES

(1, '192.168.1.10') ,(2, '192.168.1.11 '), (3, '111.118.1.12'), (4, '111.118.1.13)”)

cursorObj.execute(

"DELETE FROM mytable mt

WHERE EXISTS (

SELECT \* FROM mytable ex

WHERE ex.ipaddress = mt.ipaddress

AND ex.id < mt.id”))

con.commit()

return jsonify({'success': True })

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = True)