JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -3 EXAMINATION- 2023

B.Tech-III Semester (IT)

COURSE CODE (CREDITS): 18B11CI315 (3) MAX. MARKS: 35

COURSE NAME: Python Programming with Raspberry PI

COURSE INSTRUCTORS: Vikas Baghel MAX. TIME: 2 Hours

***Note:** (a) All questions are compulsory.*

(b) Marks are indicated against each question in square brackets.

(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems.

Q1. a) What are the differences between lists and tuples in Python? [1] **[CO1]**

b) Describe the steps involved in setting up a Raspberry Pi for the first time, from initial boot to configuration. [2]

c) Write a Python script to blink an LED connected to GPIO pins on a Raspberry Pi. [2]

Q2. a) Explain the role of CSS in web development. [1] **[CO3]**

b) What Python libraries can be used to control a web browser? [2]

c) What role does the **requests** library play in Python when working with web content? Explain with example. [2]

Q3. a) What is the significance of the bind() and listen() methods in socket programming? [2] **[CO5]**

b) Explain the differences between TCP and UDP socket communication and when you might choose one over the other. [1]

c) Write a Python program that creates a simple TCP server and allows multiple clients to connect simultaneously. [2]

d) Develop a program that allows a client to send a file to a server using the Socket module in Python. [3]

Q4. a) Explain the basic steps to play audio using PyAudio in Python. [1] **[CO6]**

b) Can you outline the differences between blocking and non-blocking modes in PyAudio? [1]

c) How can you adjust the font style, size, and color when overlaying text onto the PiCamera's output? [1]

d) Write a script that captures audio and stores it as a WAV file. [3]

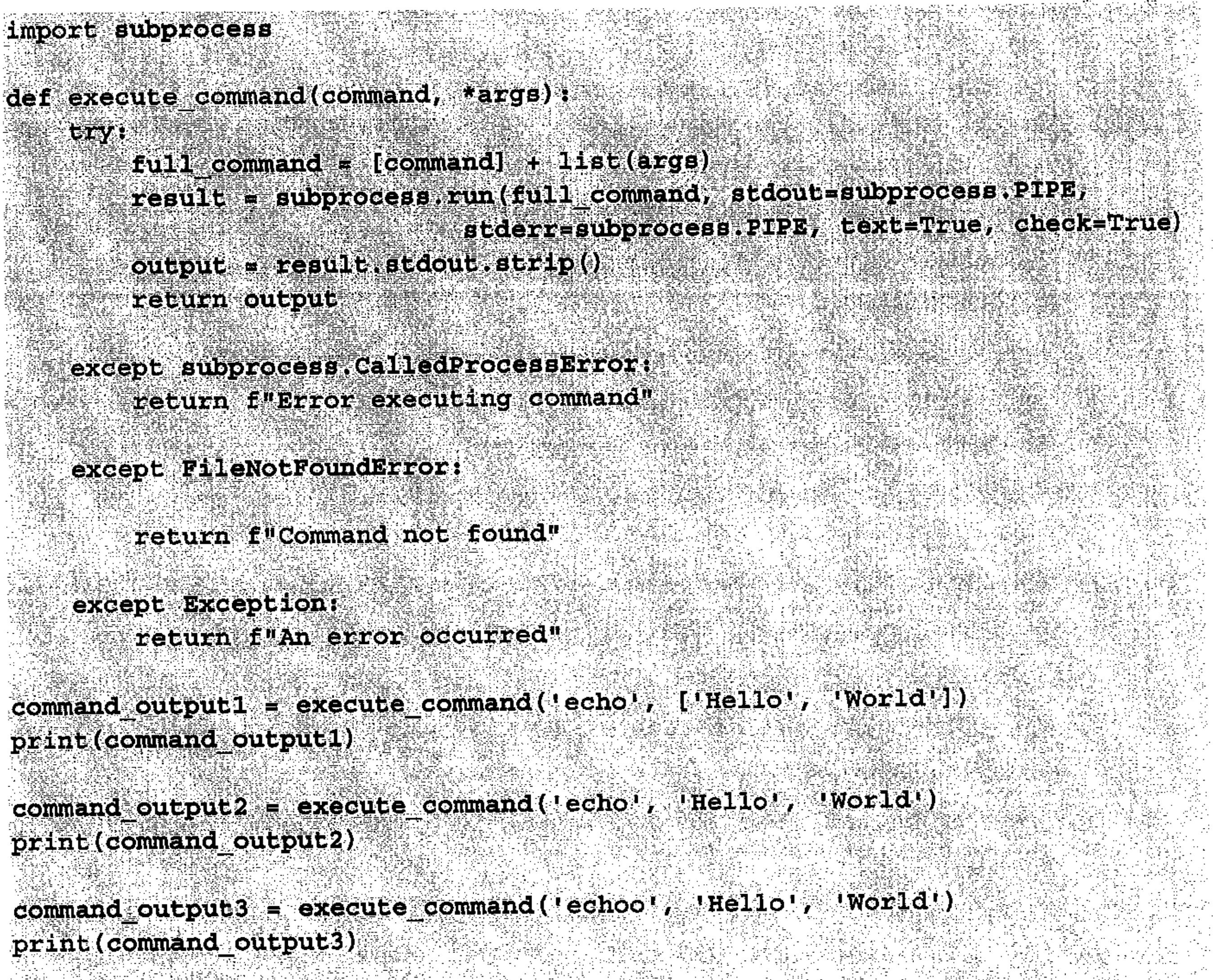
e) Implement a script that overlays a timestamp on the video feed with PiCamera, updating it dynamically. [3]

Q5. a) Explain the difference between **subprocess.run()** and **subprocess.Popen()**. [1] [CO2]

b) Write a Python script that uses subprocess to execute the **ls** command and prints the output. [2]

c) Develop a script that executes a command and handles possible errors (e.g., command not found, permission denied) using subprocess. [2]

d) Write the outputs of the program: [3]



```
import subprocess

def execute_command(command, *args):
    try:
        full_command = [command] + list(args)
        result = subprocess.run(full_command, stdout=subprocess.PIPE,
                                stderr=subprocess.PIPE, text=True, check=True)
        output = result.stdout.strip()
        return output

    except subprocess.CalledProcessError:
        return f"Error executing command"

    except FileNotFoundError:

        return f"Command not found"

    except Exception:
        return f"An error occurred"

command_output1 = execute_command('echo', ['Hello', 'World'])
print(command_output1)

command_output2 = execute_command('echo', 'Hello', 'World')
print(command_output2)

command_output3 = execute_command('echoo', 'Hello', 'World')
print(command_output3)
```