**PRACTICAL NO 1**

Create a program to send encrypted messages from sender and decrypt a message at receiver end

**Create a sender.py**

import socket

import random

def main():

# Connect to the server

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as sc:

sc.connect(('localhost', 6017))

# Get input from the user

s = input("Enter the string: ")

# Initialize variables

key = []

ct = ''

# Encrypt the message

for char in s:

j = random.randint(0, 49) # Random integer between 0 and 49

key.append(str(j)) # Append to key list

ct += chr(ord(char) + j) # Encrypt character

print(f"j={j}") # Print the random number used for each character

# Convert key list to a comma-separated string

key\_str = ','.join(key)

# Print the key and encrypted message

print(f"Key={key\_str}")

print(f"Encrypted message: {ct}")

# Send the encrypted message and key

message = f"{ct},{key\_str}"

sc.sendall(message.encode('utf-8')) # Send as bytes

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Receiver.py**

**import socket**

**def main():**

**# Create a server socket**

**with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_socket:**

**server\_socket.bind(('localhost', 6017))**

**server\_socket.listen(1) # Listen for incoming connections**

**print("Waiting for a connection...")**

**# Accept a connection**

**conn, addr = server\_socket.accept()**

**with conn:**

**print(f"Connected by {addr}")**

**# Read the incoming data**

**ct = conn.recv(1024).decode('utf-8') # Buffer size is 1024 bytes**

**s = ct.split(",") # Split the message and key**

**encrypted\_message = s[0]**

**key = list(map(int, s[1:])) # Convert key strings to integers**

**print(f"Encrypted message: {encrypted\_message}")**

**# Decrypt the message**

**pt = ''**

**for i in range(len(encrypted\_message)):**

**j = key[i]**

**pt += chr(ord(encrypted\_message[i]) - j) # Decrypt character**

**print(f"Key={j}")**

**print(f"Message from Sender: {pt}")**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**

**FIRST RUN RECEIVER THEN SENDER**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**OUTPUT**

**A screenshot of a computer

Description automatically generated**

**PRACTICAL NO 2**

import logging

# Configure the logger

logger = logging.getLogger('cfprac2')

logger.setLevel(logging.DEBUG)

# Create a file handler that logs debug and higher level messages

file\_handler = logging.FileHandler('D:/mylogfile.log', mode='a') # 'a' for append mode

file\_handler.setLevel(logging.DEBUG)

# Create a formatter and set it for the handler

formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')

file\_handler.setFormatter(formatter)

# Add the file handler to the logger

logger.addHandler(file\_handler)

# Log messages

logger.info("My first log")

logger.info("This is CFL Prac 2")

print("Logging complete. Check 'D:/mylogfile.log' for details.")

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**PRACTICAL NO 3**

**WWRITE A PROGRAM FOR SEACHING FILE IN GIVEN DIRECTORY**

import os

def main():

# Get directory from user

dir\_path = input("Enter Directory: ")

# Get first letter of file from user

first\_letter = input("Enter first letter of file: ")

try:

# List files in the directory

files = os.listdir(dir\_path)

# Filter files that start with the specified letter

filtered\_files = [file for file in files if file.startswith(first\_letter)]

if not filtered\_files:

print("No files found starting with that letter.")

else:

for filename in filtered\_files:

print(filename)

except FileNotFoundError:

print("Either dir does not exist or is not a directory")

except Exception as e:

print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

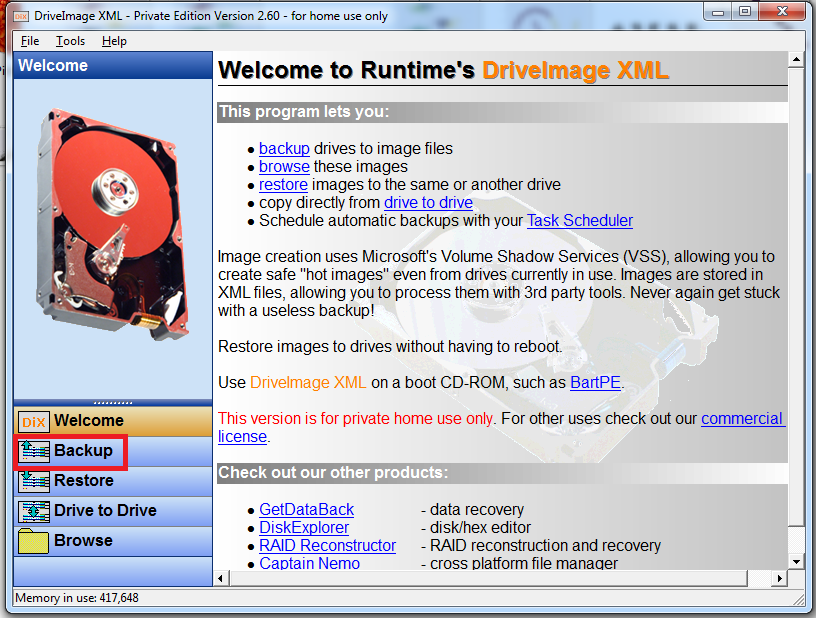
Description automatically generated

PRACTICAL NO 4

WRITE A PROGRAM TO SEARCH A PARTICULAR WORD IN A FILE

PRATICAL NO 5

**USE DRIVE IMAGE XML TO IMAGE A HARD DRIVE**



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Practical 7**

**Aim:-** Create forensic images of digital devices from volatile data such as memory using imager for computer system.

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**Practical 8**

**Aim:-** Recovering and inspecting deleted files.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer error message

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated