TP1 : Programmation Réseau Année : 2022/2023

A ) ----------------------------------------------------------------------------------------------------------------------------------------------------------

1. Ajouter des commentaires aux instructions pour clarifier le travail de chacune d’elles, et corrigé les erreurs.
2. Tester les deux applications .

# **CLIENT 2 tcp**

import socket

def client\_program():

host = socket.gethostname()

port = 5000

client\_socket = socket.socket()

client\_socket.connect((host, port))

message = input(" -> ") # take input

while message.lower().strip() != 'bye':

client\_socket.send(message.encode())

data = client\_socket.recv(1024).decode()

print('Received from server: ' + data)

message = input(" -> ")

client\_socket.close()

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

client\_program()

# **CLIENT 1 tcp**

import socket

T\_PORT = 5006

TCP\_IP = '127.0.0.1'

BUF\_SIZE = 1024

MSG = "Hello karl"

# create a socket object name 'k'

k = socket.socket (socket.AF\_INET, socket.SOCK\_STREAM)

k.connect((TCP\_IP, T\_PORT))

k.send(MSG)

data = k.recv(BUF\_SIZE)

k.close

# **SERVEUR 2 tcp**

import socket

def server\_program():

host = socket.gethostname()

port = 5000 # initiate port no above 1024

server\_socket = socket.socket()

server\_socket.bind((host, port))

server\_socket.listen(2)

conn, address = server\_socket.accept()

print("Connection from: " + str(address))

while True:

data = conn.recv(1024).decode()

if not data:

break

print("from connected user: " + str(data))

data = input(' -> ')

conn.send(data.encode())

conn.close()

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

server\_program()

# **SERVEUR 1 tcp**

import socket

T\_PORT = 60

TCP\_IP = '127.0.0.1'

BUF\_SIZE = 30

# create a socket object name 'k'

k = socket.socket (socket.AF\_INET, socket.SOCK\_STREAM)

k.bind((TCP\_IP, T\_PORT))

k.listen(1)

con, addr = k.accept()

print ('Connection Address is: ' , addr)

while True :

data = con.recv(BUF\_SIZE)

if not data:

break

print ("Received data", data)

con.send(data)

con.close()

B ) ----------------------------------------------------------------------------------------------------------------------------------------------

Ecrire un scanner de ports en Python.

#import pyfiglet

import sys

import socket

from datetime import datetime

#ascii\_banner = pyfiglet.figlet\_format("PORT SCANNER")

#print(ascii\_banner)

# Defining a target

if len(sys.argv) == 2:

# translate hostname to IPv4

target = socket.gethostbyname(sys.argv[1])

else:

print("Invalid amount of Argument")

# Add Banner

print("-" \* 50)

print("Scanning Target: " + target)

print("Scanning started at:" + str(datetime.now()))

print("-" \* 50)

try:

# will scan ports between 1 to 65,535

for port in range(1,65535):

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

socket.setdefaulttimeout(1)

# returns an error indicator

result = s.connect\_ex((target,port))

if result ==0:

print("Port {} is open".format(port))

s.close()

except KeyboardInterrupt:

print("\n Exiting Program !!!!")

sys.exit()

except socket.gaierror:

print("\n Hostname Could Not Be Resolved !!!!")

sys.exit()

except socket.error:

print("\ Server not responding !!!!")

sys.exit()