# Name: keyur Patel

# Batch : A3

# Roll No : 16010421073

# Exp-6(Inlab)

# Code:

## Server Side:

import socket import random

HOST = "127.0.0.1"

PORT = 65432

def primitive\_root(p):

number\_list = [numbers for numbers in range(1,p)] prime\_number\_list = list(sympy.primerange(2,p))

for prime\_num in prime\_number\_list: checking\_number\_list = []

for num in number\_list: checking\_number\_list.append((prime\_num \*\* (num -1)) % p) checking\_number\_list.sort()

if number\_list == checking\_number\_list: return prime\_num

def key\_calculation(value, random\_num, p): return ((value \*\* random\_num) % p)

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s: s.bind((HOST, PORT))

s.listen()

conn, addr = s.accept() with conn:

print(f"Connected by {addr}") p = conn.recv(1024)

p = int(p.decode()) g = conn.recv(1024) g = int(g.decode())

print(f"Received p ({p}) and g ({g}).") Ra = conn.recv(1024)

Ra = int(Ra.decode()) print(f"Received Ra ({Ra}).")

b = random.randint(1, p - 1) Rb = key\_calculation(g, b, p)

conn.send(str(Rb).encode()) print(f"Sent Rb ({Rb}).")

Kab = key\_calculation(Ra, b, p) print("Shared key on server side: " , Kab)

## Client Side:

import socket import sympy import random

HOST = "127.0.0.1"

PORT = 65432

def primitive\_root(p):

number\_list = [numbers for numbers in range(1,p)] prime\_number\_list = list(sympy.primerange(2,p))

for prime\_num in prime\_number\_list: checking\_number\_list = []

for num in number\_list: checking\_number\_list.append((prime\_num \*\* (num -1)) % p) checking\_number\_list.sort()

if number\_list == checking\_number\_list: return prime\_num

def key\_calculation(value, random\_num, p): return ((value \*\* random\_num) % p)

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s: s.connect((HOST, PORT))

p = int(input("Please enter a prime number: ")) g = primitive\_root(p)

s.send(str(p).encode())

s.send(str(g).encode()) print(f"Sent p ({p}) and g ({g}).")

a = random.randint(1, p - 1) Ra = key\_calculation(g, a, p)

s.send(str(Ra).encode())

print(f"Sent Ra ({Ra}).") Rb = s.recv(1024)

Rb = int(Rb.decode()) print(f"Received Rb ({Rb}).") Kab = key\_calculation(Rb, a, p)

print("Shared key on client side: " , Kab)

# Output:

## Server Side:

**ClientSide:**

