EXP. NO: 11 Implementation of Ping Program.
10/10/25

AIM: To implement a program that sends ICMP (Ping) to a host & displays the response time.

CODE:

```
import os
import platform
import subprocess

def ping(host = "google.com", count = 4):
    """
    Simple wrapper around the system ping command.
    Works on Windows, Linux & macOS.
    """
    param = "-n" if platform.system().lower() == "windows" else "-c"
    command = ["ping", param, str(count), host]

    try:
        output = subprocess.check_output(command, universal_newlines = True)
        print(output)
    except Exception as e:
        print(f"Ping failed: {e}")

if __name__ == "__main__":
    ping("8.8.8.8")
```

SAMPLE INPUT & OUTPUT:

```
ping("8.8.8.8")
```

PING 8.8.8.8 (8.8.8.8) 56 (84) bytes of data.

64 bytes from 8.8.8.8: icmp_seq = 1 ttl = 118 time = 15.2 ms

64 bytes from 8.8.8.8: icmp_seq = 2 ttl = 118 time = 14.8 ms

64 bytes from 8.8.8.8: icmp_seq = 3 ttl = 118 time = 15.1 ms

64 bytes from 8.8.8.8: icmp_seq = 4 ttl = 118 time = 14.9 ms

4 packets transmitted, 4 received, 0% packet loss

rtt min/avg/max/mdev = 14.8/15.0/15.2/0.2 ms

RESULT: The program successfully pinged 8.8.8.8 & displayed the RTT for each packet, demonstrating network connectivity using Python.

10/10 2/3/4/25

```python
import os
import platform
import subprocess

def ping(host="google.com", count=4):
    param = "-n" if platform.system().lower() == "windows" else "-c"
    command = ["ping", param, str(count), host]
    try:
        output = subprocess.check_output(command, universal_newlines=True)
        print(output)
    except Exception as e:
        print(f"Ping failed: {e}")

if __name__ == "__main__":
    ping("8.8.8.8")
```

```
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=6ms TTL=119
Reply from 8.8.8.8: bytes=32 time=9ms TTL=119
Reply from 8.8.8.8: bytes=32 time=5ms TTL=119
Reply from 8.8.8.8: bytes=32 time=9ms TTL=119

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 9ms, Average = 7ms
```