Experiment 13: Implement Your Own Ping Program

AIM:

To implement a program that sends ICMP Echo Requests (ping) to a host and 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, and 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") # Google DNS by default
```

INPUT:

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

OUTPUT:

```
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** and displayed the **round-trip time** for each packet, demonstrating network connectivity using Python.