

### Spiegazione

- Per fare questo progetto ho utilizzato Visual Studio Code e ho denominato il file `network_monitor`.
- Inizialmente implemento i moduli `os`, `platform` e `time`.
  - 'Os' per interagire con il sistema operativo.
  - 'Platform' per la piattaforma che usiamo.
  - 'Time' per utilizzarlo successivamente nel delay delle richieste di ping.

```
1 import os
2 import platform
3 import time
```

- Definisco la funzione `ping_host()`. In questa funzione prendo l'indirizzo IP degli host in input e mando un pacchetto ICMP all'host specificato utilizzando il comando 'ping' e questo mi da un return true se risponde, false se non risponde.

```
def ping_host(host):
    """
    Mando un pacchetto ICMP all'host che ritorna true se risponde false se non risponde
    """
    if platform.system().lower() == 'windows':
        ping_cmd = f"ping -n 1 {host}"
    else:
        ping_cmd = f"ping -c 1 {host}"
    response = os.system(ping_cmd)
    return response == 0
```

- Definisco ora la funzione `monitor_hosts()`. In questa funzione prendo sempre l'indirizzo IP dell'host in input e ogni 1 secondo controllo la disponibilità usando la funzione `ping_host`, precedentemente definita. In base alla disponibilità l'output può essere 'Online' oppure 'Offline'.

```
def monitor_hosts(hosts):
    """
    Monitoraggio disponibilità degli host e stampo il loro stato
    """
    while True:
        for host in hosts:
            if ping_host(host):
                print(f"{host}: Online")
            else:
                print(f"{host}: Offline")
        time.sleep(1) # rifaccio il check dopo 1 sec.
```

- Definisco infine la funzione **main()**. Con questa funzione, invece, prendo la lista degli host che vorrei controllare, che deve essere inserita dall'user dividendoli per una virgola; la funzione poi li dividerà con `split()`. Usando la funzione `monitor_hosts` precedentemente definita, monitoro gli host.

```
def main():
    """
    Faccio il get degli host da monitorare e monitoro

    """
    hosts = input("Immettere la lista degli host da monitorare dividendo per virgola: ")
    hosts = [host.strip() for host in hosts.split(",")]
    print("Monitoraggio hosts:")
    for host in hosts:
        print(f" - {host}")
    monitor_hosts(hosts)
```

- Infine, per controllare gli errori e le eccezioni ho implementato **try: except()**. In questo modo controllo, ad esempio, le `KeyboardInterrupt`, ovvero quando l'user preme `ctrl + c` per stoppare, oppure altri eventuali errori che ci possano essere e verrà poi stampato l'error message..

```
try:
    main()
except KeyboardInterrupt:
    print("Monitoraggio stoppato da user") #usando ctrl + c
except Exception as e:
    print(f"Errore: {e}")
```

### Come fare run

- Aprire il file su VS code e fare Run.
- Successivamente nel terminale risulterà scritto: “ Immettere la lista degli host da monitorare dividendo per virgola:”

```
Immettere la lista degli host da monitorare dividendo per virgola: 
```

- Possiamo mettere alcuni host come ad esempio “ google.com, example.com, 192.168.1.1” e premiamo invio.

```
Immettere la lista degli host da monitorare dividendo per virgola: google.com, example.com, 192.168.1.1
```

- Avremo la possibilità di vedere la situazione degli host che abbiamo inserito. Verrà rinviato il ping ogni 1 secondo.

Monitoraggio hosts:

- google.com
- example.com
- 192.168.1.1

Esecuzione di Ping google.com [216.58.204.238] con 32 byte di dati:

Risposta da 216.58.204.238: byte=32 durata=17ms TTL=116

Statistiche Ping per 216.58.204.238:

Pacchetti: Trasmessi = 1, Ricevuti = 1,

Persi = 0 (0% persi),

Tempo approssimativo percorsi andata/ritorno in millisecondi:

Minimo = 17ms, Massimo = 17ms, Medio = 17ms

google.com: Online

- Infine premendo CTRL + C possiamo stoppare tutto e vedremo la scritta "Monitoraggio stoppato da user".

```
192.168.1.1: Online
Monitoraggio stoppato da user
Monitoraggio stoppato da user
```