

# Fondamenti di Comunicazione e Internet

Laboratorio - 14 Giugno 2021

<b>Cognome</b>	
<b>Nome</b>	
<b>Matricola</b>	

## Parte 1 - Programmazione Socket (3 punti)

La seguente applicazione simula l'utilizzo di un servizio di acquisti online. L'utente deve prima inserire la località da cui acquista, così da acquisire la lista di servizi disponibili. Successivamente, indica il servizio e sceglie il prodotto da acquistare, che verrà consegnato se disponibile. Si risponda alle seguenti domande.

**NB:** Un dizionario in Python è una struttura del tipo:

```
dict = {"FirstKey" : [1,2], "SecondKey" : [3,4]}
```

"FirstKey" e "SecondKey" sono le chiavi del dizionario, ottenibili come lista tramite il comando:

```
list(dict.keys())
```

Scelta la chiave, si accede all'oggetto corrispondente, es. :

```
dict["FirstKey"] (ritorna la lista [1,2])
```

### Script Client:

```
from socket import *
serverName = 'localhost'
serverPort = 12001
clientSocket = socket(AF_INET, SOCK_DGRAM)
clientSocket.settimeout(5)
message = input('Inserisci una Location:')
clientSocket.sendto(message.encode('utf-8'), (serverName, serverPort))
reply, serverAddress = clientSocket.recvfrom(2048)
reply = reply.decode('utf-8')
if reply not in 'KO':
    print('Scegli un servizio tra quelli disponibili: ', reply)
    indice = input('Inserisci servizio:')
    clientSocket.sendto(indice.encode('utf-8'), (serverName, serverPort))
    reply2, serverAddress = clientSocket.recvfrom(2048)
    reply2 = reply2.decode('utf-8')
    if reply2 not in 'KO':
        print('Prodotti Disponibili: ', reply2)
        indice2 = input('Inserisci Prodotto:')
        clientSocket.sendto(indice2.encode('utf-8'), (serverName, serverPort))
        reply3, serverAddress = clientSocket.recvfrom(2048)
        reply3 = reply3.decode('utf-8')
        if reply3 not in 'KO':
            print('Prodotto in Arrivo!')
        else:
            print('Prodotto non Disponibile (Non in Listino o Fine Stock), chiusura sessione')
    else:
        print('Servizio non presente in questa località, chiusura sessione')
else:
    print('Nessun Servizio in questa località, chiusura sessione')
clientSocket.close()
```

### Script Server:

```
from socket import *
servizi = {"Milano": ["VendoLibri", "MobilePhones"],
          "Roma": ["MobilePhones", "ArticoliCasa"],
          "Napoli": ["ArticoliCasa", "VendoLibri"]}
prodotti = {"VendoLibri": ["HarryPotter", "GameOfThrones"],
            "MobilePhones": ["Iphone", "HuaweiMate"],
            "ArticoliCasa": ["Sedia", "Tavolo"]}

disponibilità = {"VendoLibri": [3,0],
                 "MobilePhones": [2,1],
                 "ArticoliCasa": [0,5]}

serverName = 'localhost'
serverPort = 12001
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
while 1:
    location, clientAddress = serverSocket.recvfrom(2048)
    location = location.decode('utf-8')
    status = 'KO'
    if location in list(servizi.keys()):
        serverSocket.sendto(str(servizi[location]).encode('utf-8'),
                           clientAddress)
    srv, clientAddress = serverSocket.recvfrom(2048)
    srv = str(srv.decode('utf-8'))
    if srv in list(servizi[location]):
        print(prodotti[srv])
        serverSocket.sendto(str(prodotti[srv]).encode('utf-8'),
                           clientAddress)
    prd, clientAddress = serverSocket.recvfrom(2048)
    prd = str(prd.decode('utf-8'))
    if prd in list(prodotti[srv]):
        index = prodotti[srv].index(prd)
        if disponibilità[srv][index] > 0:
            status = 'OK'
            disponibilità[srv][index] -= 1
    serverSocket.sendto(status.encode('utf-8'),
                       clientAddress)
```

### Domande:

**Q1) Che tipo di Protocollo di Trasporto viene utilizzato? (1 punto)**

**UDP**

**Q2) Completare il codice del: (1 punto)**

**Client:**

**indice.encode('utf-8'), (serverName, serverPort)**

**Server:**

**serverSocket.recvfrom(2048)**

**Q3) Scrivere l'output del client quando viene ordinata una "Sedia" a "Roma": (1 punto)**

Inserisci una Location:Roma

Scegli un servizio tra quelli disponibili: ['MobilePhones', 'ArticoliCasa']

Inserisci servizio:ArticoliCasa

Prodotti Disponibili: ['Sedia', 'Tavolo']

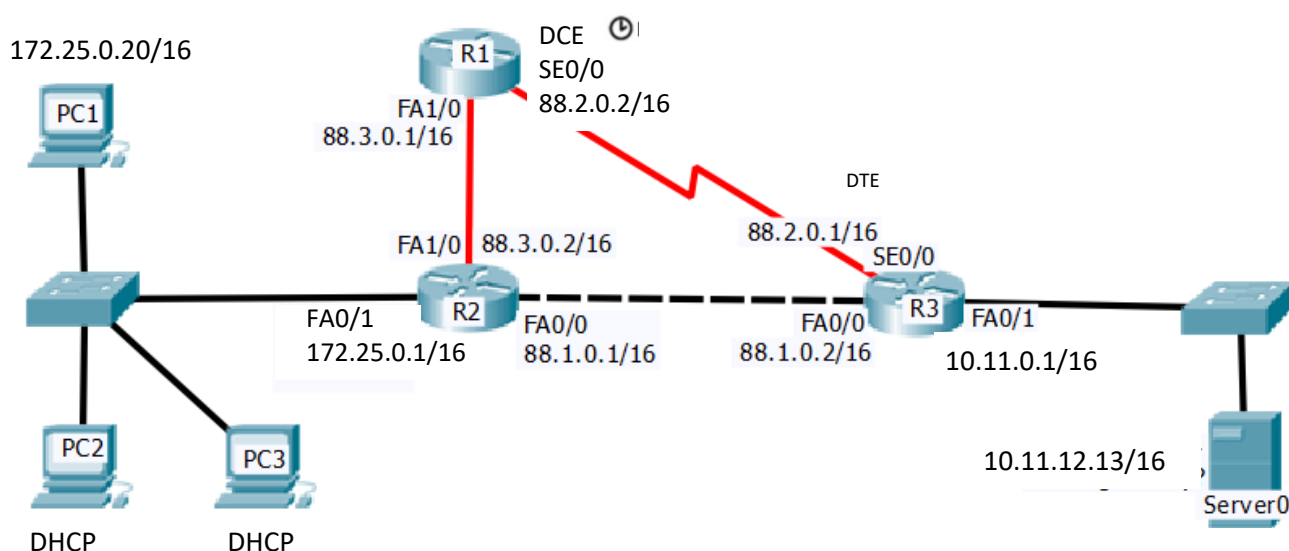
Inserisci Prodotto:Sedia

Prodotto non Disponibile (Non in Listino o Fine Stock), chiusura sessione

## **Parte 2 - Packet Tracer (3 punti)**

Si consideri la rete in figura e il suo piano di indirizzamento

**NB: Scrivere in modo esplicito la modalità del router in cui deve essere eseguito ogni comando**



**Q1) Configurare ed attivare le interfacce con indirizzi IP pubblici del router R2 assegnando i corretti indirizzi IP sulla base delle informazioni riportate in figura. (1 punto)**

```
R2> enable
R2# configure terminal
R2(config)# interface FA1/0
R2(config-if)# ip address 88.3.0.2 255.255.0.0
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# interface FA0/0
R2(config-if)# ip address 88.1.0.1 255.255.0.0
R2(config-if)# no shutdown
```

**Q2) Configurare RIPv2 nel router R3 per le sole reti PUBBLICHE. (1 punto)**

```
R3# configure terminal
R3(config)# router rip
R3(config-router)# version 2
R3(config-router)# network 88.1.0.0
R3(config-router)# network 88.2.0.0
```

**Q3) Configurare R2 come server DHCP in modo tale che agli host PC2 e PC3 venga assegnato automaticamente una configurazione IP. Usare POLIMI come nome del pool. (1 punto)**

```
R2(config)# ip dhcp pool POLIMI
R2(dhcp-config)# default-route 172.25.0.1
R2(dhcp-config)# network 172.25.0.0 255.255.0.0
R2(dhcp-config)# exit
R2(config)# ip dhcp excluded-address 172.25.0.20
```