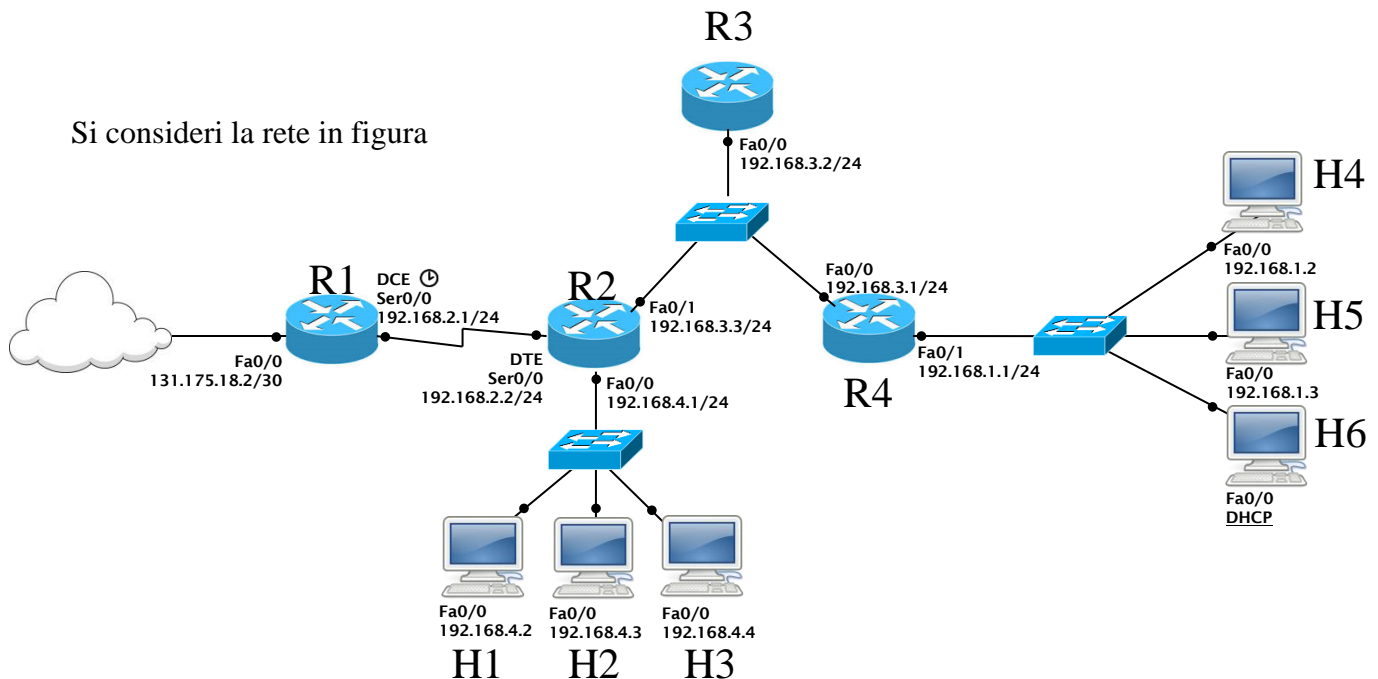


Cognome	
Nome	
Matricola	

Q1) Packet Tracer (4 punti)

Si consideri la rete in figura



Attenzione:

- Indirizzi IP e gateway sono già stati configurati per tutti gli host
- Il router R4 non è stato ancora configurato
- Indicare sempre prima del comando il prompt visualizzato dal sistema, prestando attenzione alla modalità di partenza in ciascuna richiesta

1) Configurare gli indirizzi e attivare entrambe le interfacce del router **R4**

```

R4> enable
R4# configure terminal
R4(config)# interface Fa0/0
R4(config-if)# ip address 192.168.3.1 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# exit
R4(config)# interface Fa0/1
R4(config-if)# ip address 192.168.1.1 255.255.255.0
R4(config-if)# no shutdown
    
```

- 2) Configurare il routing statico sul router **R2** in modo che si possano raggiungere tutte le subnet private e la rete Internet, minimizzando il numero di regole necessarie.
Inoltre salvare le modifiche in modo che siano disponibili al prossimo riavvio del dispositivo.

```
R2# conf t
R2(config)# ip route 192.168.1.0 255.255.255.0
192.168.3.1
R2(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1
R2(config)# exit
R2# copy running-config startup-config
```

Q2) Socket Programming (2 punti)

Si vuole scrivere un'applicazione client/server UDP per conteggiare il numero di vocali presenti in una stringa.

Il client chiede all'utente di inserire una stringa, il server risponde indicando il numero di vocali presenti nella stringa (sia maiuscole che minuscole). **Hint:** `y.count(x)` conta quante volte appare l'elemento `x` nella lista `y`.

UDP client

```
from socket import *
serverName = 'localhost'
serverPort = 9999

clientSocket = socket(AF_INET, SOCK_DGRAM)
clientSocket.settimeout(5)

message = raw_input('Inserisci una frase:')
clientSocket.sendto(message, (serverName, serverPort) )

try:
    reply, serverAddress = clientSocket.recvfrom(2048)
    print reply
except error, v:
    print "Il server non ha risposto entro il timeout..."
finally:
    clientSocket.close()
```

UDP server

```
from socket import *

serverPort = 9999
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))

print 'Server pronto a ricevere!'

vocali = ['A', 'E', 'I', 'O', 'U']

while 1:
    messaggio, clientAddress = serverSocket.recvfrom(2048)

    num = 0
    for voc in vocali:
        num = num + messaggio.count(voc)
    risposta = "Il messaggio contiene " + str(num) + " vocali."
    serverSocket.sendto(risposta, clientAddress)
```

Completare lo script "UDP client" date le seguenti specifiche:

- Utilizzare indirizzi IPv4
- Time-out in ricezione: 5 secondi.
- Lunghezza buffer di ricezione: 2048 byte.