**Laboratorio – 18/02/2021**

|  |  |
| --- | --- |
| **Cognome** | **BRAVO** |
| **Nome** | **STUDENTE** |
| **Matricola** | **SOLUZIONE** |

**Parte I – Programmazione Socket (3 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 server**  from socket import \*  serverPort = 16590  **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(1024)**  messaggio = messaggio.decode('utf-8').upper()  num = 0  for voc in vocali:  num = num + messaggio.count(voc)  risposta = "Il messaggio contiene "+str(num)+" vocali."  serverSocket.sendto(risposta.encode('utf-8'), clientAddress) |
| **UDP client**  from socket import \*  serverName = 'localhost'  serverPort = 16590  **clientSocket = socket(AF\_INET, SOCK\_DGRAM)**  **clientSocket.settimeout(3)**  message = input('Inserisci una parola (senza caratteri speciali):')  clientSocket.sendto(message.encode('utf-8'), **(serverName, serverPort)**)  try:  reply, serverAddress = **clientSocket.recvfrom(1024)**  print(reply.decode('utf-8'))  except:  print("Il server non ha risposto entro il timeout...")  finally:  clientSocket.close() |

**Q1) (vedi codice**) Completare gli script "UDP client" e "UDP server" date le seguenti specifiche:

* Utilizzare indirizzi IPv4 **(0.5 punti)**
* Time-out in ricezione (lato client): 3 secondi **(0.5 punti)**
* Lunghezza buffer di ricezione: 1024 byte **(0.5 punti)**

**Q2)** Con quale messaggio risponde il server se il client invia la stringa *"LabOratoRio"* ?

Come fa il Server a riconoscere sia lettere maiuscole che minuscole?

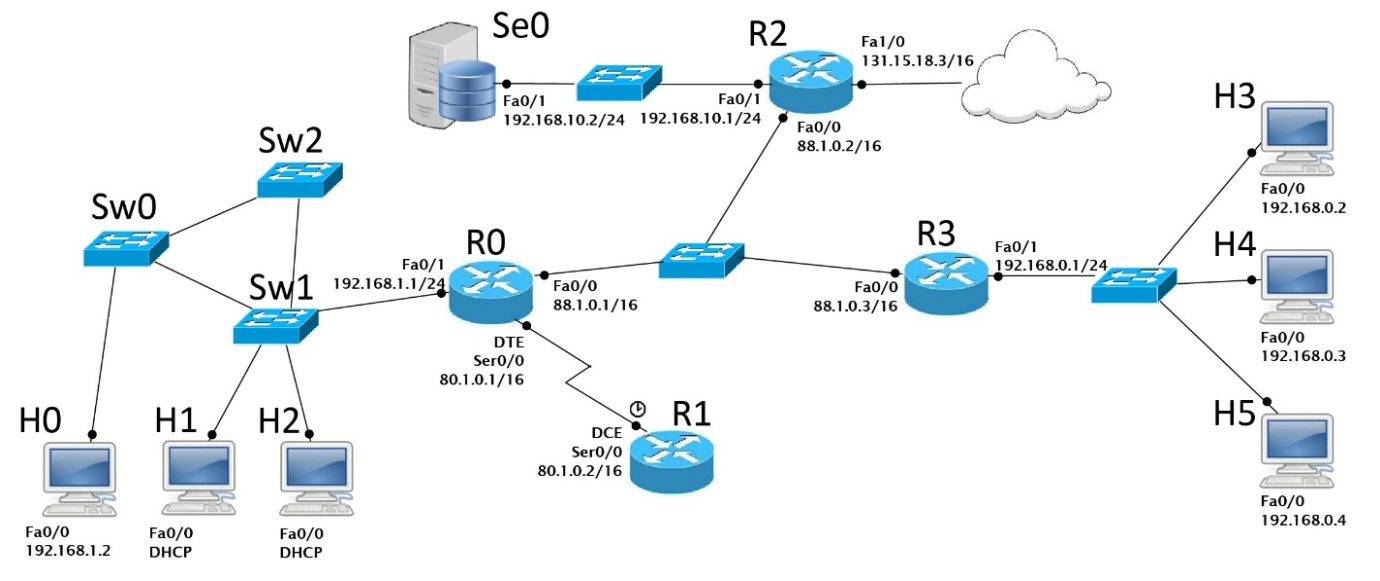
**(1.5 punti)**

**La risposta del server è: Il messaggio contiene 6 vocali**

**Il server è in grado di riconoscere vocali maiuscole e minuscole grazie all’utilizzo della funzione upper() che rende ogni lettera del messaggio del client maiuscola prima del controllo sul numero di vocali.**

**Parte II – Packet Tracer (3 punti)**

**Si consideri la rete in figura**

****

**Attenzione:**

* **Indirizzi IP e gateway sono già stati configurati per i 6 host.**
* **Le interfacce dei router R1, R2, e R3 sono già state configurate ed attivate come in figura.**
* **Le reti /24 sono reti private**
* **Indicare sempre prima del comando il prompt visualizzato dal sistema, prestando attenzione alla modalità di partenza in ciascuna richiesta**

1. **Configurare ed attivare le interfacce Fa0/0 e Fa0/1 del router R0 come in figura. (1 punto)**

R0> **enable**

**R0# configure terminal**

**R0 (config)# interface Fa0/0**

**R0 (config-if)# ip address 88.1.0.1 255.255.0.0**

**R0 (config-if)# no shutdown**

**R0 (config-if)# exit**

**R0 (config)# interface Fa0/1**

**R0 (config-if)# ip address 192.168.1.1 255.255.255.0**

**R0 (config-if)# no shutdown**

**R0 (config-if)# exit**

1. **Abilitare il protocollo RIPv2 sul router R2. (1 punto)**

R2> **enable**

**R2# configure terminal**

**R2 (config)# router rip**

**R2 (config-router)# version 2**

**R2 (config-router)# network 192.168.10.0**

**R2 (config-router)# network 131.15.0.0**

**R2 (config-router)# network 88.1.0.0**

1. **Configurare R0 come server DHCP in modo tale che agli host che si collegano alla sottorete (H1 e H2) venga assegnata automaticamente una configurazione IP. (1 punto)**

R0(config)# **ip dhcp pool *pool\_name***

**R0(dhcp-config)# default-router 192.168.1.1**

**R0(dhcp-config)# network 192.168.1.0 255.255.255.0**