Laboratorio - 3 Febbraio 2020

Cognome	
Nome	
Matricola	

Q1) Socket Programming (6 punti)

Il codice sotto riportato rappresenta l'applicazione "Eat-It": il server è posto in ascolto sulla porta 12000 all'indirizzo 131.175.21.43 e periodicamente attende che un utente (Client) invii le proprie coordinate GPS, così da rispondere con l'elenco dei luoghi di ristoro presenti nelle vicinanze. Successivamente, se lo desidera, l'utente sceglie uno tra i luoghi disponibili e comunica al server l'orario di prenotazione.

Q1 Completare il codice mancante nel Server e nel Client (2 punti)

Q2 Si riporti ciò che appare sul terminale client nelle due situazioni in cui il client: (3 punti)

A) 1) Decide di voler prenotare, 2) Sceglie uno dei locali proposti, 3) Sceglie come orario di prenotazione le 21

B) 1) Decide di voler prenotare, 2) Sceglie un locale non presente nella lista

Q3 Cosa appare sul terminale server nelle due situazioni considerate (A, B), supponendo che l'indirizzo Ip del client sia 131.175.21.47 e la porta sia la 56947? (1 punto)

Q1) Vedere sotto

Q2)
A) EatIt invia le tue coordinate al server...
Luoghi disponibili vicino a te:
['Sushi', 'FastFood']
Vuoi Prenotare? si
Scegli una soluzione nella lista proposta: Sushi
A che ora vuoi mangiare? 21
Prenotato alle 21 in Susho. Affrettati!

B) EatIt invia le tue coordinate al server...
Luoghi disponibili vicino a te:
['Sushi', 'FastFood']
Vuoi Prenotare? si
Scegli una soluzione nella lista proposta: < qualsiasi stringa diversa da Sushi, FastFood >
Soluzione non disponibile, chiusura sessione.

Q3) Connection from: ('131.175.21.47', 56947)

#Script client

```
from socket import *

def get_GPS_coord():
    return (7,14)

serverName = 'localhost'
serverPort = 12000
clSock = socket(AF_INET, SOCK_STREAM)
clSock.connect((serverName, serverPort))
print('Eat-It invia le tue coordinate al server…')
# Invia coordinate GPS al server
(latitudine,longitudine) = get_GPS_coord()
clSock.send(str(latitudine).encode('utf-8'))
message = clSock.recv(2)
if message.decode('utf-8') == 'ok':
    clSock.send(str(longitudine).encode('utf-8'))
# Legge dal server i luoghi disponibili nelle vicinanze
numero_scelte =  int(clSock.recv(100).decode('utf-8'))
lista_scelte = []
# Richiede, uno alla volta, i luoghi disponibili
print('Luoghi disponibili vicino a te:')
for i in range(numero_scelte):
    clSock.send('Next'.encode('utf-8'))
    scelta = clSock.recv(100).decode('utf-8')
    lista_scelte.append(scelta)
print(lista_scelte)
decisione = input('Vuoi Prenotare? ')
if decisione == 'si':
# Selezione una delle scelte disponibili
    prenotazione = input('Scegli una soluzione nella lista
proposta: ')
    clSock.send(prenotazione.encode('utf-8'))
    message = clSock.recv(100).decode('utf-8')
    if message == 'ok':
        orario = input('A che ora vuoi mangiare? ')
        clSock.send(str(orario).encode('utf-8'))
        message = clSock.recv(100).decode('utf-8')
        if message == 'OK':
            print('Prenotato alle '+orario+' in '+prenotazione
+'. Affrettati!')
    else:
        print('Soluzione non disponibile, chiusura
sessione.')
elif decisione == 'no':
    clSock.send('no'.encode('utf-8'))
    print('Ciao!')
else:
    print('Scelta errata, chiusura sessione!')
clSock.close()
```

#Script server

```
from socket import *

def get_list_from_GPS(latitudine, longitudine):
    return ["Sushi", "FastFood"]

serverPort = 12000
servSock = socket(AF_INET, SOCK_STREAM)
servSock.bind(('', serverPort))
servSock.listen(5)

while True:
    clSock, clAddr= servSock.accept()
    print("Connection from: ", clAddr)
    # Riceve dal client le coordinate GPS
    lat = clSock.recv(100).decode('utf-8')
    clSock.send("ok".encode('utf-8'))
    lng = clSock.recv(100).decode('utf-8')
    # Ricava la lista di luoghi disponibili
    places = get_list_from_GPS(lat, lng)
    clSock.send(str(len(places)).encode('utf-8'))
    for place in places:
        message = clSock.recv(1024).decode('utf-8')
        if message == 'Next':
            clSock.send(place.encode('utf-8'))
    selezione = clSock.recv(1024).decode('utf-8')
    if str(selezione) in places:
        clSock.send('ok'.encode('utf-8'))
        ora = str(clSock.recv(1024).decode('utf-8'))
        clSock.send('OK'.encode('utf-8'))
    elif str(selezione) == 'no'
        print('')
    else:
        clSock.send('Errore'.encode('utf-8'))
    clSock.close()
```