# Laboratorio

(6 punti)

Il codice sotto riportato rappresenta l’applicazione “Art-It”: il server è posto in ascolto sulla porta 15000 all’indirizzo 131.171.25.42 e periodicamente attende che un utente (Client) invii le proprie coordinate GPS, così da rispondere con l’elenco delle mostre d’arte presenti nelle vicinanze. Successivamente, se lo desidera, l’utente sceglie una tra le mostre disponibili e comunica al server l’orario di prenotazione.

**Q1** Completare il codice mancante nel Server e nel Client  
**Q2** Si riporti ciò che appare sul terminale client nelle due situazioni in cui il client:

1. 1) Decide di voler prenotare, 2) Sceglie la mostra ‘Picasso’, 3) Sceglie come orario di prenotazione le 16
2. 1) Decide di voler prenotare, 2) Sceglie la mostra ‘Giotto’

**Q3** Cosa appare sul terminale server nelle due situazioni considerate (A, B), supponendo che l’indirizzo IP e il numero di porta lato client siano rispettivamente 131.171.25.47 e 56946?

**Q1)** Vedere sotto

**Q2)**

A) Art-It invia le tue coordinate al server...

Mostre disponibili vicino a te:

[‘Picasso’, ‘Monet’]

Vuoi Prenotare? si

Scegli una soluzione nella lista proposta: Picasso

A che ora vuoi prenotare? 16

Prenotato alle 16 per Picasso. Affrettati!

B) Art-It invia le tue coordinate al server...

Mostre disponibili vicino a te:

[‘Picasso’, ‘Monet’]

Vuoi Prenotare? si

Scegli una soluzione nella lista proposta: ‘Giotto’

Mostra non disponibile, chiusura sessione.

**Q3)** Connection from: ('131.171.25.47', 56946)

**#Script client**

from socket import \*

def get\_GPS\_coord():

return (8,16)

serverName = 'localhost'

serverPort = 15000

**clSock = socket(AF\_INET, SOCK\_STREAM)**

clSock.connect((serverName, serverPort))

print('Art-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 le mostre disponibili nelle vicinanze

numero\_scelte = int(clSock.recv(100).decode('utf-8'))

lista\_scelte = []

# Richiede, uno alla volta, le mostre disponibili

print('Mostre 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 prenotare? ')

clSock.send(str(orario).encode('utf-8'))

message = clSock.recv(100).decode('utf-8')

if message == 'OK':

print('Prenotato alle '+orario+' per '+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 ["Picasso", "Monet"]

serverPort = 15000

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()