

Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Lënda: Rrjetat Kompjuterike – 2017

Projekti 1:

Dizajnimi Klient – Server duke përdorur Socketët (ang. Sockets)

UDP dhe TCP

Studentja: Monika Behluli

Asistenti : Msc. Edon Mustafa

Data : 18/04/2017

Permbajtja

**1.Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara3**

**2.Hyrje4**

2.1 Pershkrim I shkurter rreth projektit4

2.2 Si punon arkitektura klient/server?4

2.3 Pershkrimi I klientit dhe serverit4

**3.Realizimi I TCP Klientit dhe serverit permes kodit** ………………………………………………………………………………**5**

3.1 Realizimi I TCPserver…………………..…………………………………………………………………………………………….5

3.2 Realizimi I TCPclient………………………………………………………………………………………………………………….6

**4.Realizimi I UDP Klientit dhe serverit permes kodit** …………………………………………………………………………….7

4.1 Realizimi I UDPserver…………………..…………………………………………………………………………………………..7

4.2 Realizimi I UDPclient………………………………………………………………………………………………………………..8.

**5.Pershkrimi I metodave dhe rezultatet pas tetstimit**.……………………………………………………………………….11

5.1 Pershkrimi i metodave pa hyrje11

5.1.1 Metoda IP………………………………………………………………………………………………………………………….11

5.1.2 Metoda PORT……………………………………………………………………………………………………………………12

5.1.3 Metoda HOST……………………………………………………………………………………………………………………13

5.1.4 Metoda TIME…………………………………………………………………………………………………………………….15

5.1.5 Metoda KENO……………………………………………………………………………………………………………………16

5.2 Pershkrimi I metodave me hyrje……………………………………………………………………………………………………17

5.2.1 Metoda ZANORE…………………………………………………………………………………………………………………17

5.2.2 Metoda PRINTO………………………………………………………………………………………………………………….18

5.2.3 Metoda FAKTORIEL…………………………………………………………………………………………………………..19

5.2.4 Metoda KONVERTO…………………………………………………………………………………………………………….20

**6.Pershkrimi I metodave personale dhe rezultatet pas testimit**.…………………………………………………….23

6.1 Metoda QUIZI……………..……………………………………………………………………………………………………….23

6.2 Metoda NUMRAT PERFEKT ………………………………………………………………………………………………….27

6.3 Metoda REVERSE………………………………………………………………………………………………………………….27

Veglat e perdorura, versionet e tyre dhe lista e metodave te implementuara

Sistemi operativ:

* Windows 10

Veglat e perdorura dhe versionet e tyre:

* Microsoft Visual Studio 2015 Ultimate
* Python 2.7

Libarite e Python:

* Socket
* Random
* Dat**e**Time
* Cprofile
* Decimal
* Tkinter

Lista e metodave te implementuara:

* IP
* PORT
* ZANORE
* PRINTO
* HOST
* TIME
* KENO
* FAKTORIEL
* KONVERTO

Lista e metodave personale te implementuara:

* COLORGAME
* THJESHTE

## 2.Hyrje

# 2.1 Pershkrim I shkurter rreth projektit

Projekti perbehet nga sistemi programimi I klienti dhe serverit te cilet komunikojne permes FIEK Protokollit.

Komunikimi ndermjet klientit dhe serverit behen permes socket-ave te cilat mundesojne komunikim e tyre ne portin default 9000 ose nje port te vecante te cilin mund ta caktoje klienti.

FIEK Protokolli eshte nje protokoll I thjeshte I komunikimit I cili per baze ka fjalen kyqe ne fillim te kerkesen e cila percakton metoden te cilen do ta kthej server. Ajo ka dy versione:

1. FIEK-TCP
2. FIEK-UDP

Ky protokoll ju pergjigjet vetem ketyre kerkesave: IP, PORT, ZANORE, PRINTO, HOST, TIME, KENO, FAKTORIEL, KONVERTO te cilat dergohen nga klienti ne server dhe per te cilat serveri kthen nga nje pergjigje te vecante, klienti se bashku me kerkesen mund te dergoje edhe input varesisht nga kerkesa.

# 2.2 Si punon arkitektura klient/server?

**Serveri** njëjtë sikurse edhe kompjuterët e tjerë është një makinë procesorike e fuqishme. Detyrat e kësaj makinë janë që të shërbejnë klientët. Serveri është console aplikacion që lidhet me klient përmes socketave dhe për secilen lidhje me klient krijon një thread të ri. Serveri dallon llojet e kërkesave që vijnë nga klienti dhe për secilin lloj ka përgjigje të veqantë. Pas seciles përgjigje mbyllet soketi.

# 2.3 Pershkrimi I klientit dhe serverit

FIEK-TCP:

Se pari vendoset nje lidhje(segment) ne mes te klientit dhe serverit permes socket-ave ne portin e caktuar(default 9000). Pastaj klienti permes command line e dergon kerkesen tek server, I cili nese kerkesa eshte valide I pergjigjet kerkeses specifike te klientit dhe me pas mbyll lidhjen me klientin. Serveri eshte ne gjendje te lidhet me me shume kliente ne te njejten kohe dhe ti kthej pergjigje klienteve ne menyre paralele.

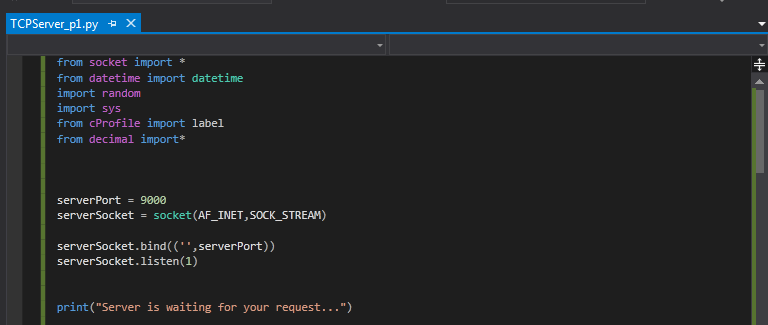
FIEK-UDP:

Ne versionin UDP te prokollit FIEK komunikimi realizohet vetem permes UDP datagram-ve, ku me c’rast nuk krijohet nje lidhje(segment) mes klientit dhe serverit por komunikimi fillon drejt. Klienti dergon kerkesen tek server permes nje UDP datagram. Pasi qe kerkesa te validohet ne server ajo poashtu kthehet tek klienti permes UDP datagram-it. Protokolli FIEK limiton qe klienti te dergon vetem nje kerkese per datagram.

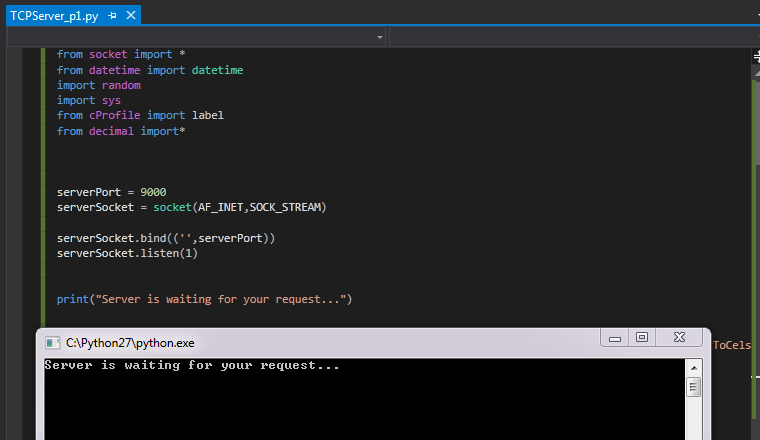
## 3. Realizimi I TCP klientit dhe serverit permes kodit

# 3.1 Realizimi I TCPserver

Se pari deklarohet variabla statike serverPort e cila perdoret gjate ekzekutimit te programit , dhe she rbejne per tu lidhur me klientin.

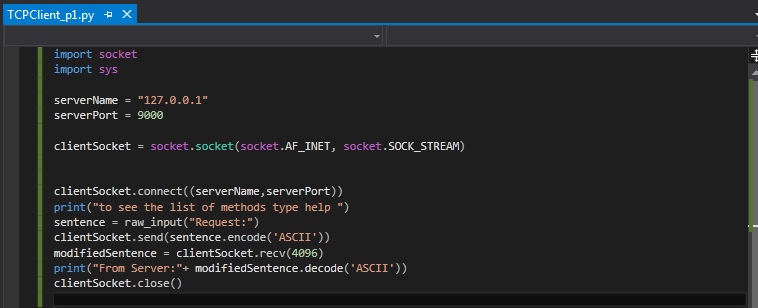


Pas ekzekutimit serveri eshte I gatshem per pranimin e kerkesave nga klienti

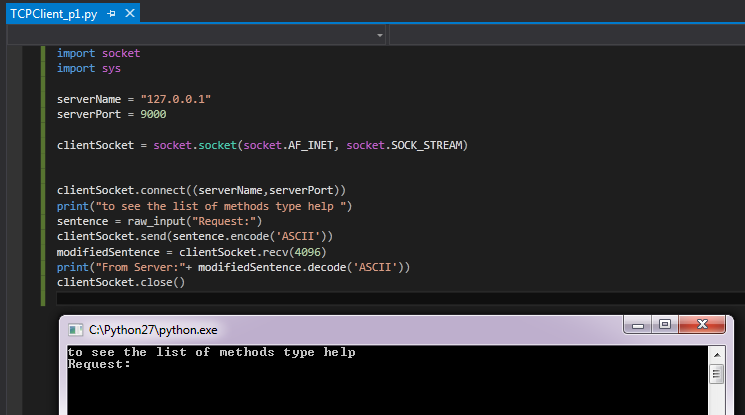


# 4.1 Realizimi I TCPclient

Ne fillim se pari deklarohen variabla serverName dhe serverPort me te cilat do te mundesojme lidhjen e klientit ne server dhe permes te ciles kryhet komunikimi.



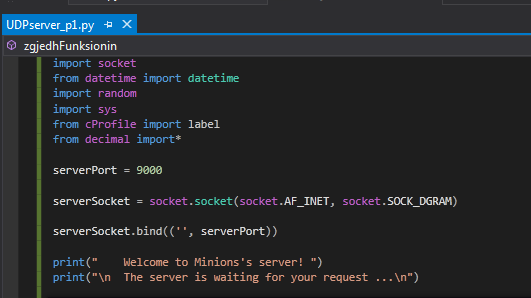
Pas ekzekutimit te TCPclient , klienti mund te beje kerkesen dhe pergjigjien e merr nga TCPserver



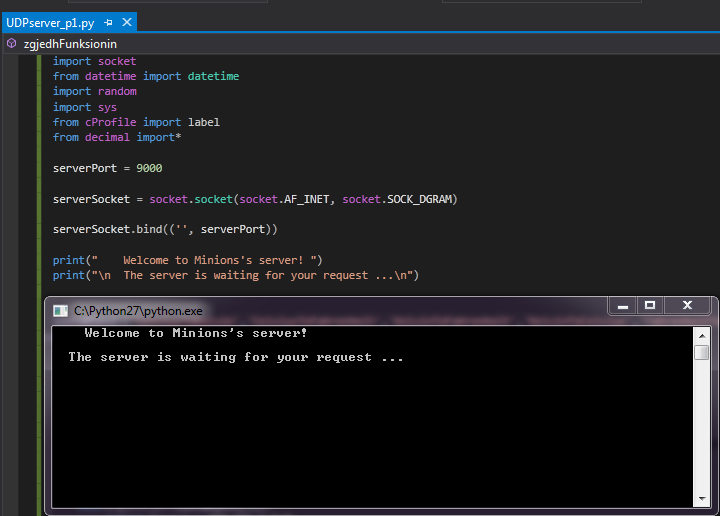
## 4.Realizimi I UDP klientit dhe serverit permes kodit

# 4.1 Realizimi I UDPserver

Se pari deklarohen variablat statike te cilat perdoren gjate ekzekutimit te programit , dhe sherbejne per tu lidhur me klientin.

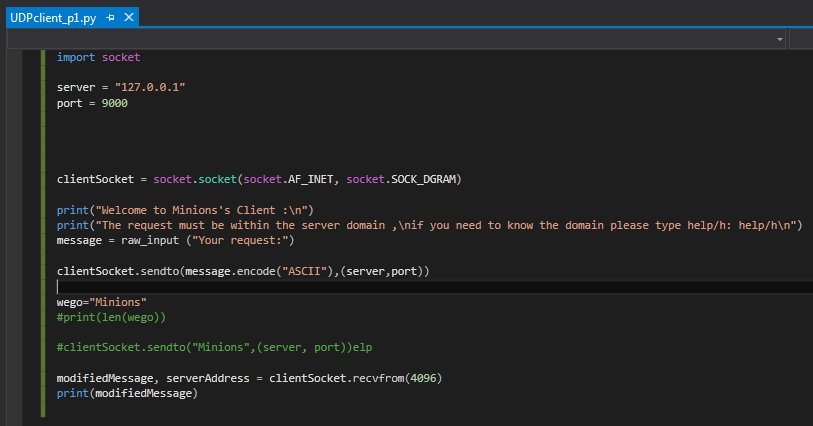


Pas ekzekutimit serveri eshte I gatshem per pranimin e kerkesave nga klienti

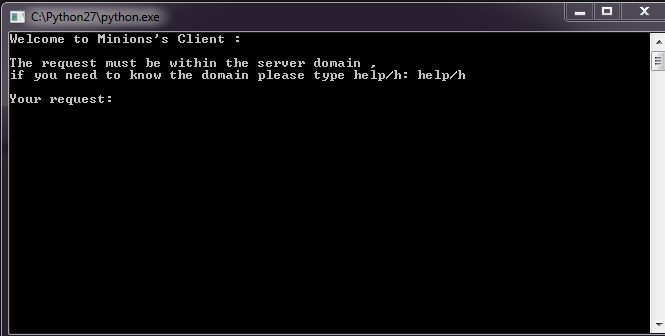


# 4.1 Realizimi I UDPclient

Ne fillim se pari deklarohen variabla server dhe port me te cilat do te mundesojme lidhjen e klientit ne server dhe permes te ciles kryhet komunikimi.



Pas ekzekutimit te UDPclient , klienti mund te beje kerkesen dhe pergjigjien e merr nga UDPserver



Per implementimin e metodave se pari deklarohet nje unaze while True: ne UDPserver, mbrenda se ciles realizohet kodi per metodat.

Fillimisht per ti mundesuar klientit njohjen e domenit eshte realizuar kerkesa HELP mbrenda unazes while True: e cila mundeson njohjen e kerkesave per te cilat UDPserver mund te kthej pergjigjie.

Realizimi permes kodit:

else: mesazhi="\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n"+\

"Lista e metodave eshte brenda domenes:\n"+\

"IP\n"+"PORT\n"+"ZANORE\n"+"PRINTO\n"+"HOST\n"+"TIME\n"+\

"KENO\n"+"Konverto\n"+"\tCelsiusToKelvin\n"+"\tCelsiusToFahrenheit\n"+\

"\tKelvinToFahrenheit\n"+"\tKelvinToCelsius\n"+"\tFahrenheitToCelsius\n"+\

"\tFahrenheitToKelvin\n"+"\tPoundToKilogram\n"+"\tKilogramToPound\n"+\

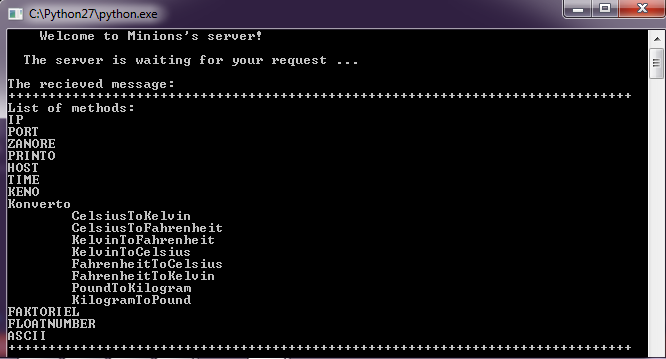
"FAKTORIEL\n"+"FLOATNUMBER\n"+"ASCII\n"+\

"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++\n"

print("Mesazhi i pranuar: " + mesazhi)

serverSocket.sendto(mesazhi, clientAddress)

Pas pranimit te kerkeses HELP ne server :



Pergjigjia qe kthehet nga serveri, pasi klienti te beje kerkesen Help :



## 5.Pershkrimi i metodave dhe rezultatet pas testimit

# 5.1 Pershkrimi i metodave pa hyrje

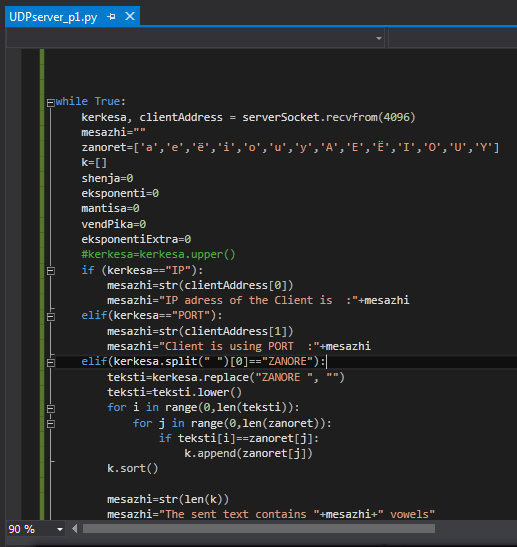
* + 1. Metoda IP

Request : IP

Pergjigjia nga serveri : Përcakton dhe kthen IP adresën e klientit në formë dhjetore me pike në një tekst mesazh p.sh. (10.10.7.251)

* Një mesazh jo me i gjate se 128 karaktere qe përmban IP adresën e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

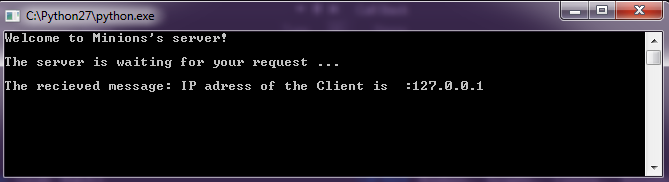
Realizimi permes kodit:

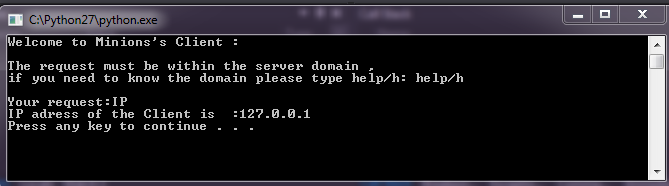


Shpjegim I kodit:Permes metodes clientAddress serveri merr ip e klientit dhe e kthen prap te ai.

Rezultatet pas testimit

Pasi klienti te beje kerkesen IP I kthehet pergjigjia nga serveri





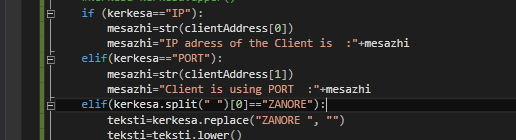
* + 1. Metoda PORT

Request : PORT

Pergjigjia nga serveri : Përcakton dhe kthen portin e klientit. (Ky duhet te jete porti I klientit dhe jo porti I serverit.)

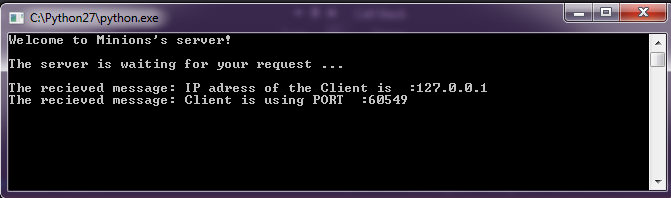
* Një mesazh jo me i gjate se 128 karaktere qe përmban IP adresën e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

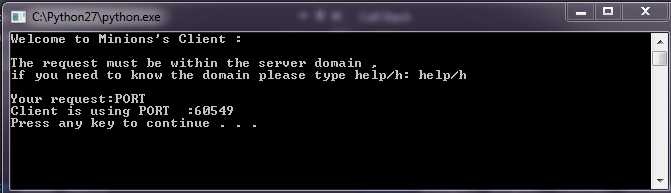
Realizimi permes kodit:



Rezultatet pas testimit

Pasi klienti te beje kerkesen PORT I kthehet pergjigjia nga serveri





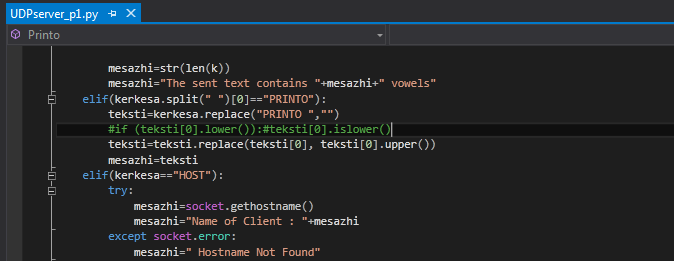
* + 1. Metoda HOST

Request : HOST

Pergjigjia nga serveri : Kërkon emrin e hostit dhe e kthen atë. Nëse emri i hostit nuk mund te përcaktohet (gjendet) atëherë duhet te kthehet një mesazh dhe te tregohet se emri i hostit nuk mund te gjendet

* Një mesazh jo me i gjate se 128 karaktere qe përmban IP adresën e klientit.
* Mesazhi nuk guxon te përmbaj NULL karakterin

Realizimi permes kodit:



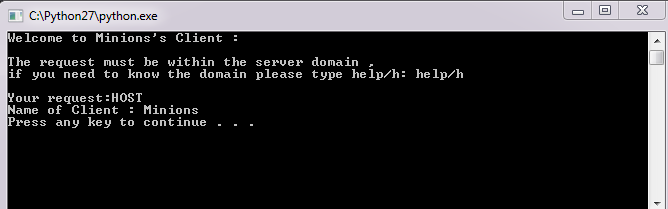
Shpjegimi I kodit:

Me ane te metodes .GetHostName() mirret emri i hostit i cili pastaj kthehet tek klienti, ne qofte se marrje e emrit te hostit ka qene e pasukseshme klienti lajmerohet qe emri i hostit nuk eshte marre.

Rezultatet pas testimit

Pasi klienti te beje kerkesen HOST I kthehet pergjigjia nga serveri





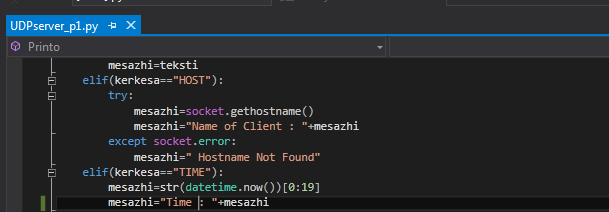
* + 1. Metoda TIME

Request : TIME

Pergjigjia nga serveri : Përcakton kohen aktuale ne server dhe e dërgon atë tek klienti si format te

lexueshme për njerëzit.

* Një mesazh jo me i gjate se 128 karaktere qe përmban kohen e serverit.
* Mesazhi nuk guxon te përmbaj NULL karakterin.

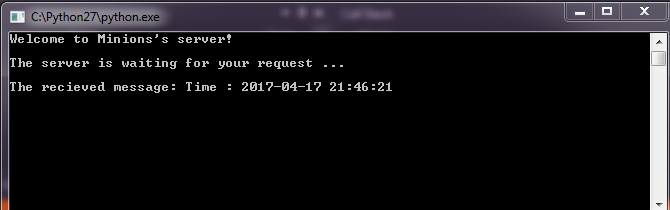
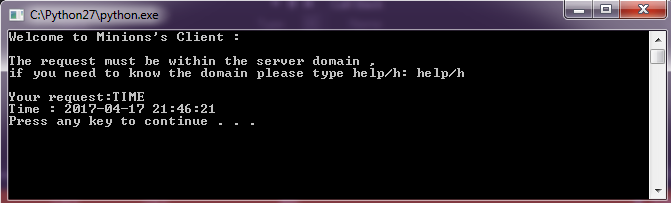
Realizimi permes kodit: 

Shpjegimi I kodit:

Permes klases DateTime mirret koha aktuale e sistemit dhe kthehet ne string ne formatin “HH:mm:ss tt

Rezultatet pas testimit

Pasi klienti te beje kerkesen TIME I kthehet pergjigjia nga serveri

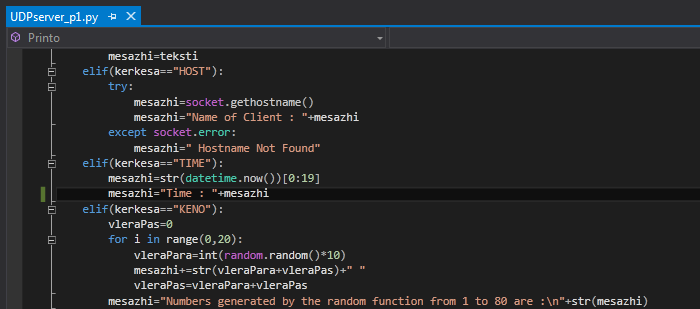
 

* + 1. Metoda KENO

Request : KENO

Pergjigjia nga serveri : Kthen 20 numra nga rangu [1,80].

* Një mesazh jo me i gjate se 128 karaktere qe përmban kohen e serverit.
* Mesazhi nuk guxon te përmbaj NULL karakterin.

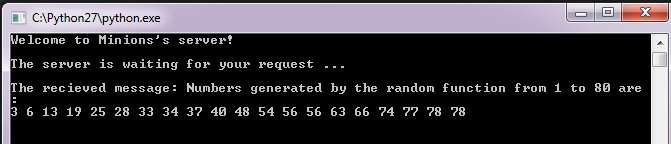
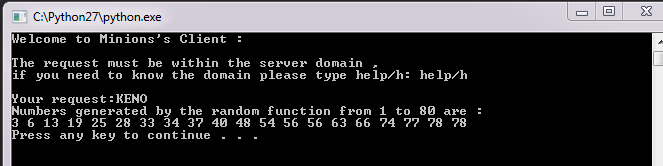
Realizimi permes kodit: 

Shpjegimi I kodit:

Klasa Random perdoret per gjenerimin e numrave te rastesishem te cilat ruhen ne nje array te tipit int dhe permes unazes for gjenerehen 20 numra random nga 1 deri 80 permes 20 cikleve . ku ne secilin cikil shikohet se a eshte numer I perseritur ne vargun e numrave te meparshem qe te mos perseritet.

Rezultatet pas testimit

Pasi klienti te beje kerkesen KENO I kthehet pergjigjia nga serveri

# 5.2 Pershkrimi i metodave me hyrje

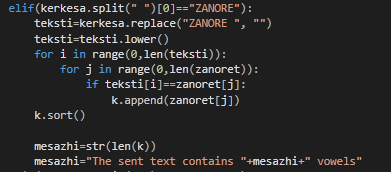
5.2.1 Metoda ZANORE

Request : ZANORE{HAPESIRE}tekst

Pergjigjia nga serveri : Gjen numrin e zanoreve ne tekst dhe kthen pergjigjien

* Një mesazh jo me i gjate se 128 karaktere qe përmban kohen e serverit.
* Mesazhi nuk guxon te përmbaj NULL karakterin.

Realizimi permes kodit:

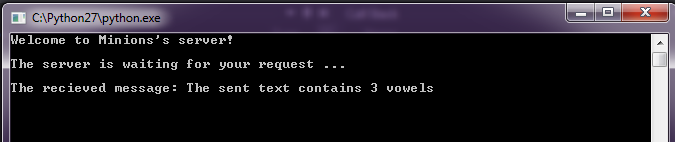
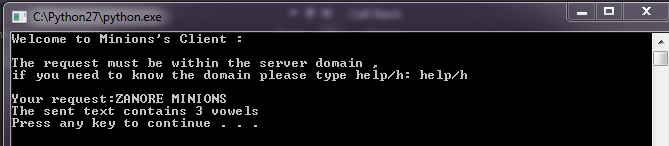
C:\Users\SINKOPA\Desktop\18009582_1757915510901747_2057160418_n.png 

Shpjegimi I kodit:

Fillimisht deklarohet vargu I zanoreve “zanoret”,pasi klienti te jep fjalen serveri e krahason nese ne fjalen e dhene nga klienti ndodhet ndonje karakter I vargut “zanoret” dhe rrit variablen per nje.Ne fund kthen numrin e zanoreve qe ndodhen ne fjalen e dhene nga klienti.

Rezultatet pas testimit

Pasi klienti te beje kerkesen ZANORE FJALE I kthehet pergjigjia nga serveri

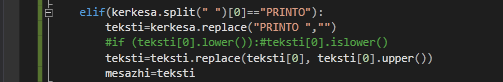
 

5.2.2 Metoda PRINTO

Request : PRINTO

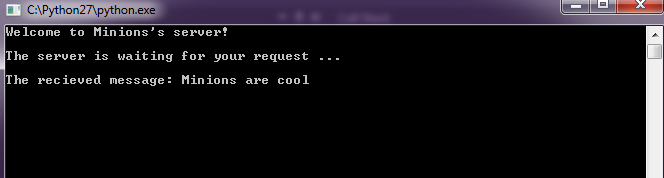
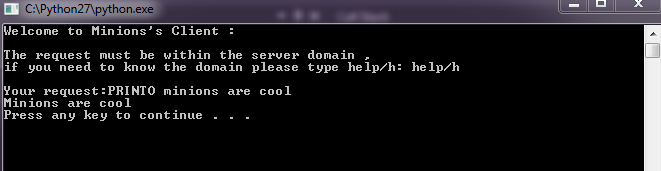
Pergjigjia nga serveri : Kthen fjalinë e shtypur ne tekst.Hapësirat ne fillim dhe ne fund te fjalisë nuk duhet te kthehen

Realizimi permes kodit:



Rezultatet pas testimit

Pasi klienti te beje kerkesen PRINTO FJALINE I kthehet pergjigjia nga serveri

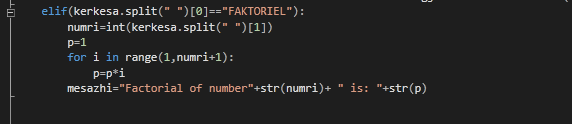
5.2.3 Metoda FAKTORIEL

Request : FAKTORIEL{HAPESIRE}numer

Pergjigjia nga serveri : Gjen faktorielin si rezultat i parametrit të dhënë hyrës. Numër i plotë psh.

FAKTORIEL 5 kthen si rezultat 120

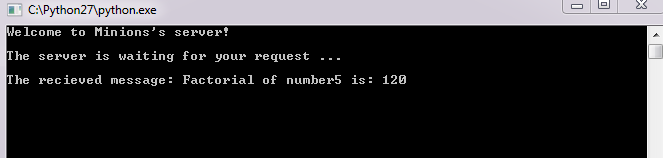
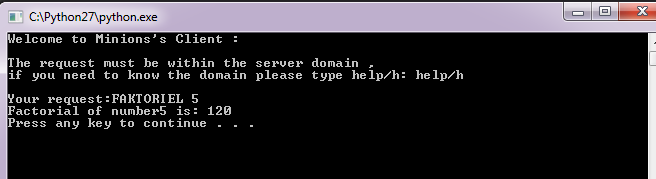
Realizimi permes kodit:



Shpjegimi I kodit:Sipas nje unaze for e cilva fillon nga 1 deri te nr qe e jep klienti dhe llogaritet si p\*I

Rezultatet pas testimit

Pasi klienti te beje kerkesen FAKTORIEL 5 I kthehet pergjigjia nga serveri

5.2.4 Metoda KONVERTO

Request : KONVERTO{HAPESIRE}opcioni{HAPESIRE}numer

Pergjigjia nga serveri : Kthen si rezultat konvertimin e opcioneve varësisht opcionit të zgjedhur.

* Lista e parametrave opcioni janë:
* CelsiusToKelvin
* CelsiusToFahrenheit
* KelvinToFahrenheit
* KelvinToCelsius
* FahrenheitToCelsius
* FahrenheitToKelvin
* PoundToKilogram
* KilogramToPound

Realizimi permes kodit:

listaNjesi=['CelsiusToKelvin','CelsiusToFahrenheit','KelvinToFahrenheit','KelvinToCelsius','FahrenheitToCelsius','FahrenheitToKelvin','PoundToKilogram','KilogramToPound']

def zgjedhFunksionin(zgjedhja, numri):

if(zgjedhja==listaNjesi[0]):

return float(numri)-273.15

elif(zgjedhja==listaNjesi[1]):

return numri \* 9/5 + 32

elif(zgjedhja==listaNjesi[2]):

return numri \* 9/5 - 459.67

elif(zgjedhja==listaNjesi[3]):

return numri - 273.15

elif(zgjedhja==listaNjesi[4]):

return (numri- 32) \* 5/9

elif(zgjedhja==listaNjesi[5]):

return (numri + 459.67) \* 5/9

elif(zgjedhja==listaNjesi[6]):

return numri \* 0.45359237

elif(zgjedhja==listaNjesi[7]):

return numri / 0.45359237

else:

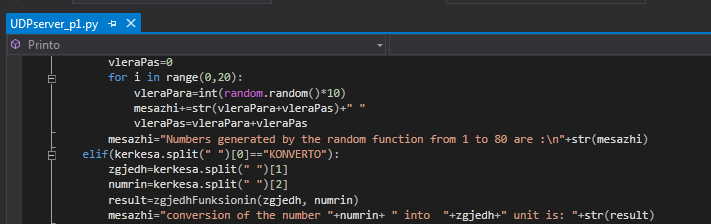
Printo()

def Printo():

print("Please choose a measure unit from the list and type a number you want to convert :")

for i in range(0,8):

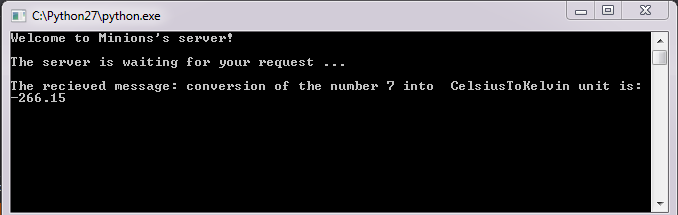
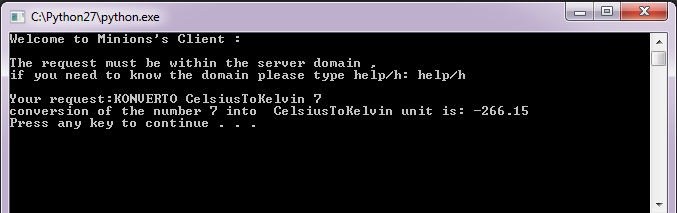
print(listaNjesi[i])



Shpjegimi I kodit:Varsishet se cilen nga mundesit e dhena per konvertim e zgjedh klienti I kthehet pergjigjia nga serveri.

Rezultatet pas testimit

Pasi klienti te beje kerkesen I kthehet pergjigjia nga serveri

## 6.Pershkrimi i metodave personale dhe rezultatet pas testimit

6.1. Metoda Kuizi

Pergjigjia nga serveri : Jep mundësi klientit të luaj një lojë kuizi të vogël në lëminë e matematikës.

Realizimi permes kodit:

import random

operatorList = [ "+", "-", "\*" ]

print 'Mire se erdhet ne Kuizin e Matematikes!'

s = 0

g = raw\_input("Ju lutem shkruani emrin : ")

for i in range(10):

nr1 = random.randint(1,10)

nr2 = random.randint(1,10)

op = random.choice(operatorList)

expression = "%d %s %d" % (nr1, op, nr2)

answer = eval(expression)

print expression, "= ???"

reply = int( raw\_input("Shkruani pergjigjejen tuaj : ") )

if (reply == answer):

print "Sakte!"

s=s+1

else:

print "Gabim! Pergjigjeja e sakte eshte", answer

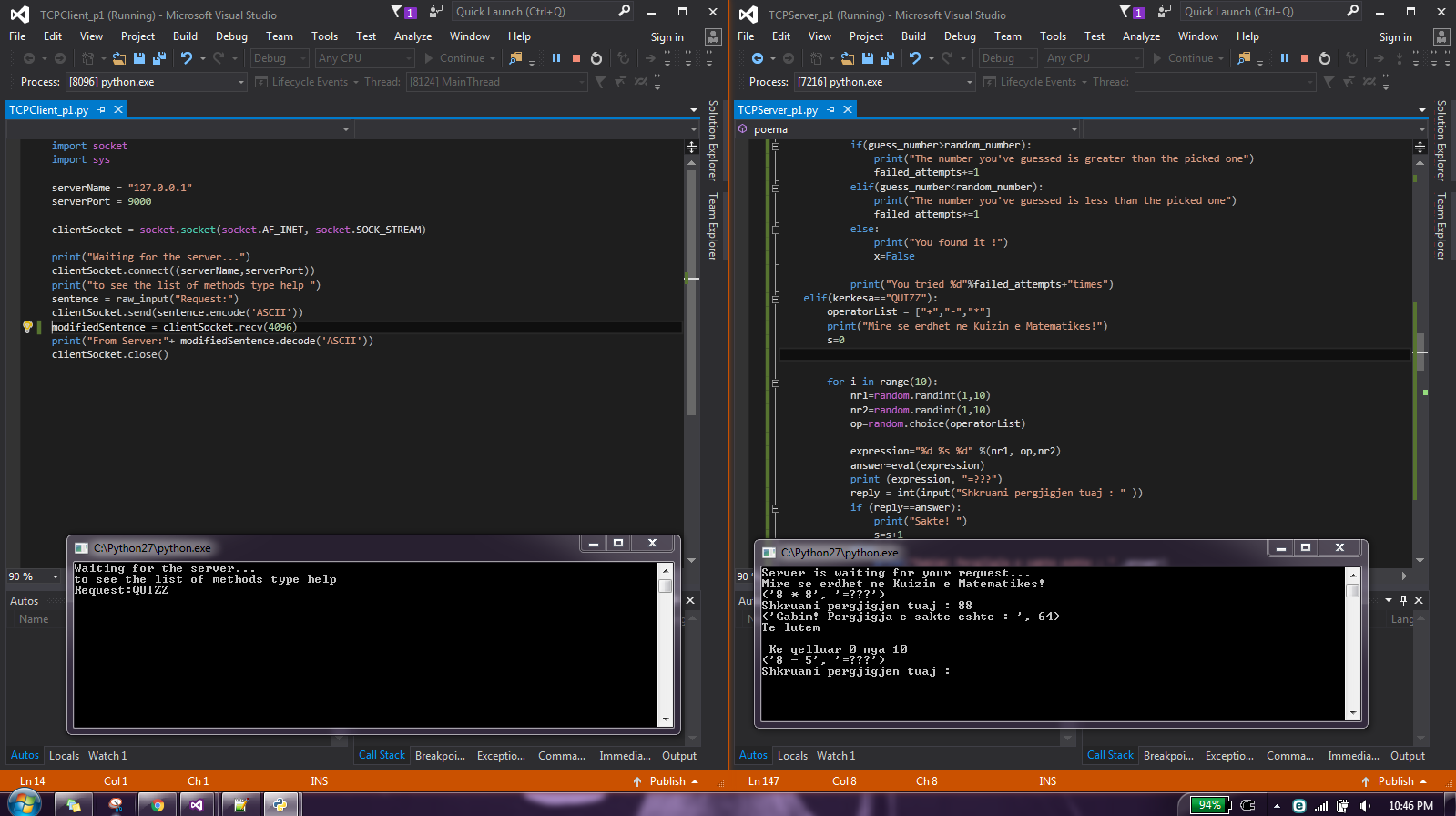
print "Te lumte"

print "\n Ke qelluar %d nga 10 " % s

Shpjegimi i kodit:

Fillimisht për këtë metodë është e nevojshme që të importohet klasa random. Është deklaruar një array i cili përmban operacionet matematikore. Inicializohet një variabël s=0 e cila do të shërbejë që të mbahet evidence për pyetjet e dhëna saktë. Numrat që do të mblidhen, zbriten apo shumëzohen do të zgjidhen random sikurse edhe vet operacioni. Varësisht nga përgjigjëja e klientit, ai do të njoftohet në qoftë se ai dha përgjigjëje të saktë apo jo. Për gjdo përgjigjëje të saktë të klientit variabla s do të rritet për një. Në fund të kuizit, klienti do të njoftohet për pytjet e qëlluara.

Rezultatet pas testimit



6.2 Metoda Numrat Perfekt

Pergjigjia nga serveri : Kthen pergjigjëje në qoftë se numri i pranuar nga klienti është numer perfekt.

Sqarim : Numri është perfekt në qoftë se shuma e plotpjesëtuesve të tij është sa vet numri.

Realizimi permes kodit:

def perfect\_number(Number):

Sum = 0

for i in range(1, Number):

if(Number % i == 0):

Sum = Sum + i

if (Sum == Number):

print(" %d eshte numer perfekt !" %Number)

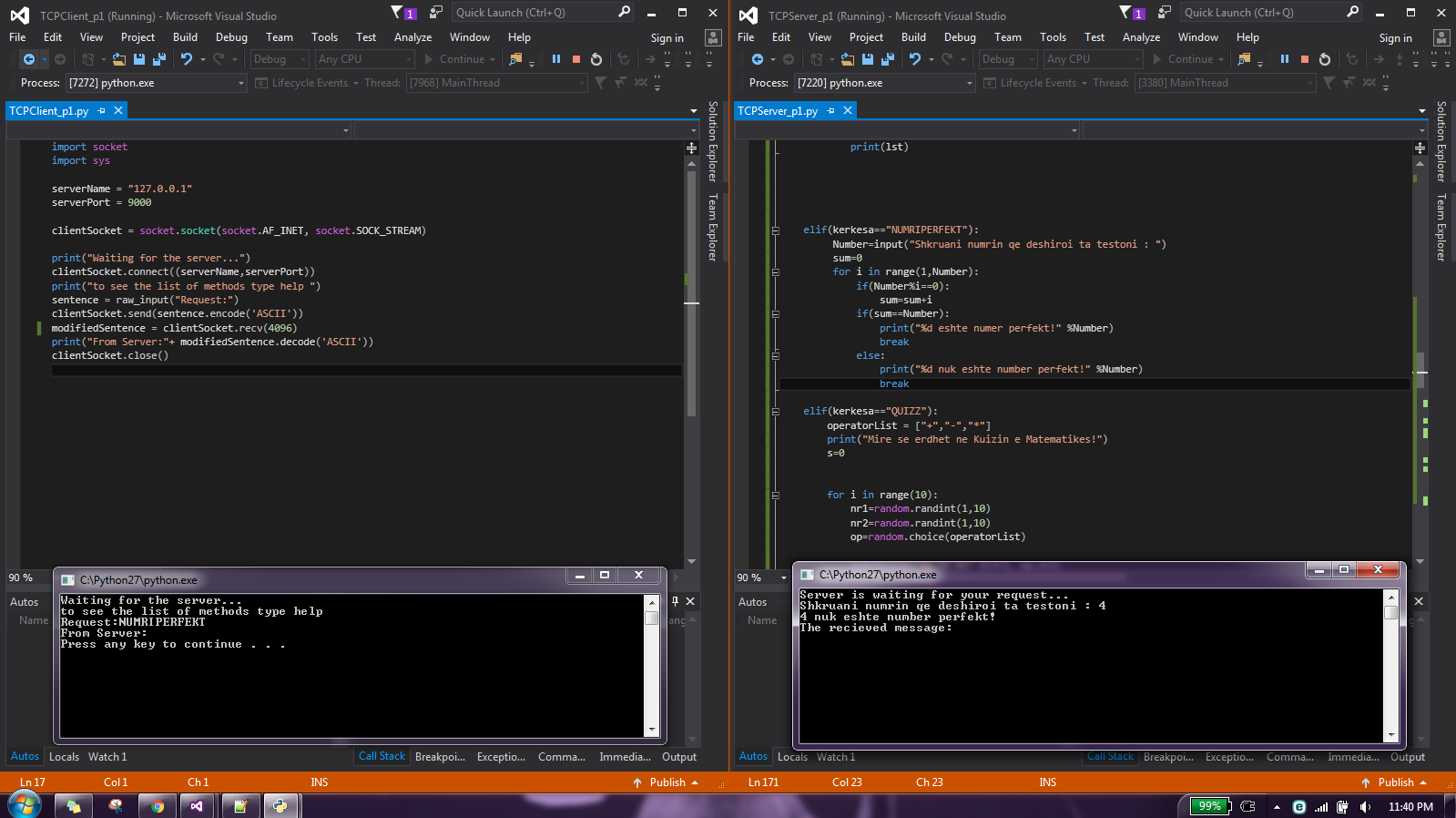
else:

print(" %d nuk eshte numer perfekt !" %Number)

Shpjegimi i kodit:

Metoda NumraPerfekt ka një parametër e cila është numri që klienti do e shtyp. Fillimisht inicializohet një variabel sum që barazohet me zero dhe me anë të një unaze e cila fillon nga 1 dhe shkon deri tek numri që ka dhënë klienti, shuma do të rritet për “i”. Pas përfundimit të unazës me anë të një kushti do të testohet në qoftë se kjo shumë është e barabartë me numrin e dhënë, dhe në qofte se po njofton klientin që numri është perfekt , përndryshe njofton që jo.

Rezultatet pas testimit



6.1. Metoda Reverse

Pergjigjia nga serveri : Fjalët e pranuara nga klienti, metoda në fjalë i përmbys.

Realizimi permes kodit:

def reverse(text):

lst = []

count = 1

for i in range(0,len(text)):

lst.append(text[len(text)-count])

count += 1

lst = ''.join(lst)

return lst

Shpjegimi i kodit:

Metoda në fjalë pranon si parametër tekstin e pranuar nga klienti. Fillimisht deklarohet një array I zbrazur dhe një variabël count=1. Me anë të një unaze lista e deklaruar në fillim mbushet me shkronjat e tekstit të marrur nga klienti duke filluar nga ato te fundit. Në fund bëhet bashkimi dhe metoda kthen fjalën e përmbysur.

Rezultatet pas testimit