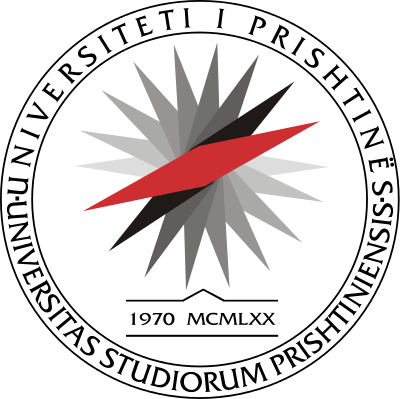
RRJETA KOMPJUTERIKE



RAPORTI I PROJEKTIT 1  
 **Punoi: Drenushë Pirraku**

**ID:160714100087**

**Prof.: Dr.Techn Blerim Rexha**

**Ass.: Msc.Haxhi Lajqi**

**Gjuha programuese: Python**

**Mjedisi: Visual Comm. Edition**

##### 2019

Table of Contents

**Parathenjie3**

Si u bë e mundur?3

TCP

UDP3

**Kuptimi i Sockets4**

Stream Sockets5

Datagram Sockets5

**Dizajni Klient-Server……………………………………....…………………………………………………………………..………6**

Pjesa kryesore(Main Part)…………………………………………………………………………………………………………………7

Pamjet vizuale………………………………………………………………………………………………………………………………8

## 

# Parathenjie 1

I gjithë ky projekt u ndërtua nga lista e kërkesave të parashturara nga mentorët, me anë të së cilave më lehtësisht mund të kuptojmë se çka është kërkuar të krijohet. Gjithashtu vlen të përmendet fakti që ështe ndërtuar perms gjuhës programuses *Python* dhe përmes veglës *Visual Studio 2019 Comm. Edition.* E cila kjo gjuhë ishte mjaft e panjohur për mua, por fal ushtrimeve dhe aplikacioneve apo faqeve online përvetësova aftësinë e programimit të një socket-i(*Socket Programming*).

#### Si u bë e mundur?

Le të flasim pak për kërkesat më detajisht, është kërkuar nga mentorët që të krijohen 4 programe, përkatësisht këto 4 programe kishin një funksion kryesor për të kryer: të lidhen mes veti dhe të komunikojnë duke shkëmbyer të dhëna.  
Programet e kërkuara përfshinin : TCP Serverin, TCP Klientin, UDP Serverin dhe UDP Klientin.

#### TCP - Transmission Control Protocol[[1]](#footnote-2)

TCP është një protokoll, një listë e rregullave përmes të cilës anembanë botës pajisjet të lidhura mes vete qoftë në lidhje lokali (LAN – Local Area Network), qoftë në lidhje të tjera komunikojnë mes vete. TCP, siç e njohim ne, mundëson transferim të besueshëm të të dhënave, në mënyrë të rregullt dhe pa gabime.

#### UDP – User Datagram Protocol[[2]](#footnote-3)

UDP nga ana tjetër është një alternativë e TCP-së e cila funksionon pak më ndryshe. UDP nga ana tjetër nuk siguron që të gjitha të dhënat tuaja do të transferohen, andaj nuk mund të themi se është reliable. Shembull: UDP e përdorim kur kemi live streaming.

Gjatë zgjidhjes së problemit ne kemi hasur edhe në shumë nocione me të cilat nuk kemi qenë të familjarizuar më herët. Ndër to mund të përmendim edhe termin **SOCKETS[[3]](#footnote-4),** nocion me të cilin kemi punuar gjatë gjithë projektit, e që mund të themi përmes të cilave edhe është realizuar ky projekt.

Kuptimi i Sockets

Socket-i në kuptimin e përgjithshëm është një pike e brendshme e cila mundëson dërgimin ose marrjen e të dhënave brenda një nyje të një rrjeti komputerik.

Socketat janë specifike për një nyje dhe ndryshe nga portat nuk mund të referohen nga nyjet në mënyre të drejtëpërdrejtë.  
Në rastin tonë të caktuar, nëse shikohet me kujdes pjesa e parë e kodit, vërehet se janë përdorur dy lloje të socketave:  
**-**Stream Sockets  
-Datagram Sockets Andaj menjëherë mund të pyesim se pse janë përdorur dy lloje të socketave? Përgjigja do të ishte shumë e thjeshtë, janë krijuar dy lloje të programeve andaj edhe na janë dashur dy lloje të socketave.  
Në vazhdim mund të pyesim se cili korrespodon cilin? Atëherë lehtësisht mund ti dallojmë, *Datagram Sockets* i korrespodon programit me UDP kurse *Stream Sockets*  përdoret për TCP. Për më detajisht le t’i qartësojmë këto dy nocione mëposhtë:

Stream Sockets

Stream Socket-i është një pikë e lidhjes se një rrjetiqë siguron një rrjedhë të orientuar në atë lidhje të rrjetës, lidhje e cila është e sekuencuar dhe gjithashtu siguron një rrjedhë të të dhënave pa pengesa, me mekanizma të definuar në mënyrë specifike të cilat krijojnë dhe prishin lidhje dhe gjithashtu janë të afta të gjejnë gabime gjatë transferimit të të dhënave.

Datagram Sockets

Nga ana tjetër një Datagram Socket, në parim kryen funksionin e njëjtë si Stream Socket-i mirëpo mënyra e funksionimit dallon.

Tek Datagram Socket-i çdo pako e dërguar ose e marrë në një Datagram Socket është e adresuar dhe orientuar (routed) në mënyrë individuale. Rregulla dhe besueshmëria, sie përmendëm edhe më herët nuk jane të garantuara, kështu që disa paketa të derëguara nga një pajisje, mund të arrijnë tek pajisja tjetër në mënyrë jo të rregullt, ose në rastin më të keq nuk mund të arrijnë fare.  
  
 Pasi të kemi kuptuar konceptet e përgjithshme rreth nocioneve me të cilat do të punojmë, hapi tjetër që vie është ndërtimi i një bazamenti të zgjidhjet, i cili mund të jetë një pseudokod apo një tabelë me të cilat ne përcaktojmë qëllimet kryesore dhe disa pika të cilat duhet të plotësohen.

# 

# Dizajni Klient-Server

Jetike është që fillimisht të vendoset një “urë” lidhëse mes këtyre dy pajisjeve(machines).

Fillimisht vendoset një port për serverin, në mëyrë që klienti kur të lidhet ta dijë saktësisht se ku do të lidhet: **serverPort = 12000**

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

Pra me anë të këtyre dy komandave janë përcaktuar dy gjëra:   
  
 Porti i serverit do të jetë 12000 dhe protokolli për komunikim do të jetë **TCP** sepse shohim se tek socket-i kemi përdorur Stream Socket për të cilën kemi spjeguar më lartë rreth funksionit të saj.  
Hapi pasues është konfigurimi i klientit në mënyrë që të krijohet lidhja. Nga ana e klientit do të kemi:  
  
**serverName = 'localhost'**

**serverPort = 12000**

**ClientSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)**

**ClientSocket.connect((serverName,serverPort))**

Rrjedhimisht nga ana e klientit janë përcaktuar disa parametra:  
Është caktuar emri i serverit ku do të lidhet që në këtë rast është “localhost” i cili si default IP e ka adresën 127.0.0.1 . Porti duhet të vendoset i njëjtë në mënyrë që lidhja të bëhet direkt me serverin që krijuam e që në fakt ka të njëjtin port.

#### Pjesa kryesore (Metodat)

Deri më tani kemi spjeguar se si është bërë e mundur lidhja mes klientit dhe serverit.  
Që tani e tutje do të merremi me pjesën kryesore të programit, e cila është pjesa e metodave.  
Metodat janë funksione, të cilat krijohen një herë për një qëllim të caktuar mirëpo mund të thirren saher që user-i të ketë nevojë për to.

Në total në ketë projekt, si detyrë e kemi pasur të krijojmë 11 metoda funksionale, secila me funksion ndryshe nga njëra tjetra.

Metodat të cilat janë kërkuar janë:  
  
IPADRESA – E cila kthen IP adresën e klientit në formë dhjetore të ndara me pikë.

NUMRIIPORTIT – E cila si rezultat kthen portin e klientit.

BASHKETINGELLORE – E cila si rezultat kthen numrin e zanoreve në një fjalë që e jep useri si input.

PRINTIMI – E cila si rezultat shtyp fjalën që useri e jep si input.

EMRIIKOMPJUTERIT – E cila gjen dhe kthen emrin e kompjuterit apo hostit.

KOHA – E cila tregon kohën dhe datën në ato moment.

LOJA – E cila si rezultat kthen 7 numra të cfarëdoshëm nga 1 deri në 49.

FIBONACCI – E cila si rezultat kthen numrin e kërkuar nga useri në sekuencën Fibonacci.

KONVERTIMI – E cila konverton disa njësi të caktuara në bazë të dëshirës së klientit.

Gjithashtu si kërkesë e projektit ka qenë që të implementohen edhe 2 metoda shtesë, të pavarura nga të tjerat, dhe të cilat nuk duhen të kenë ngjashmëri me metodat e tjera, qoftë brenda këtij projekti apo të ngjashme me metodat që kolegët tjerë i kanë krijuar.  
  
Këto dy metoda janë:  
  
FUQIA – E cila si rezultat kthen katrorin e numrit te dhene.

PRAPAFJALA – E cila fjalën e dhënë e kthen prapa.

Në vazhdim do të ndalemi tek secila metodë, do të analizojmë analogjinë e punës së tyre dhe rezultatet të cilat shfaqen.

#### Pamjet vizuale

Për sa i përket funksionalizimit dhe saktësisë të kodit më poshtë kam bashkangjitur disa pamje vizuale të zgjedhjes së metodës nga klienti dhe kthimit të pergjigjes nga server.

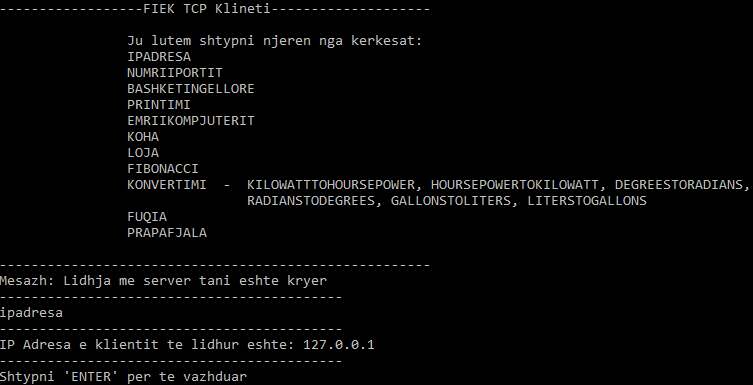


Figure 1.Kthimi I IP adreses me ane te metodes IPADRESA

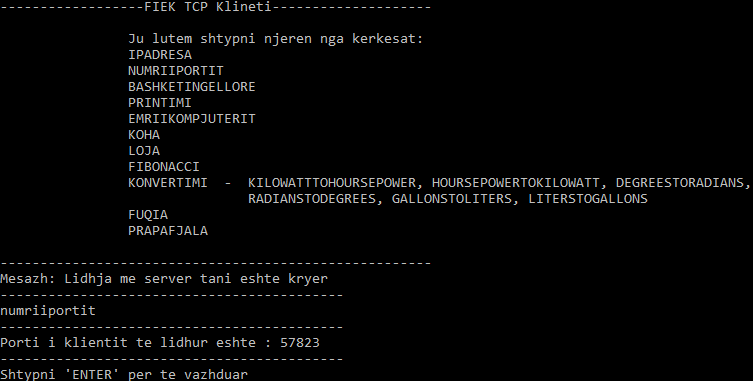


Figure 2.Kthimi i portit me anë të metodës NUMRIIPORTIT

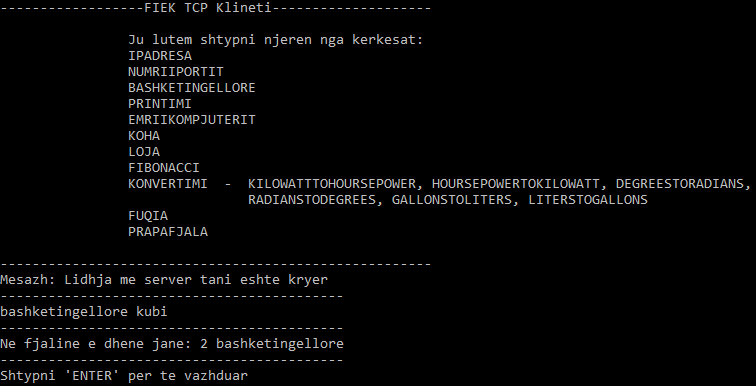


Figure3.Llogaritja e numrit të bashkëtingëlloreve përmes metodës BASHKËTINGËLLORE

#### 

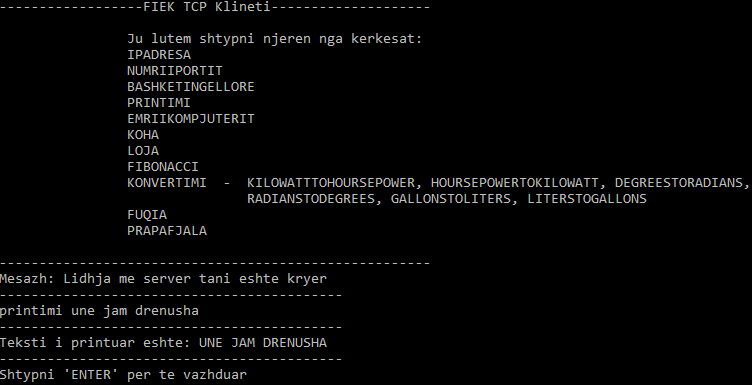


Figure 4.Kthimi i mesazhit te shkruar pa hapësira pëmes metodës PRINTIMI

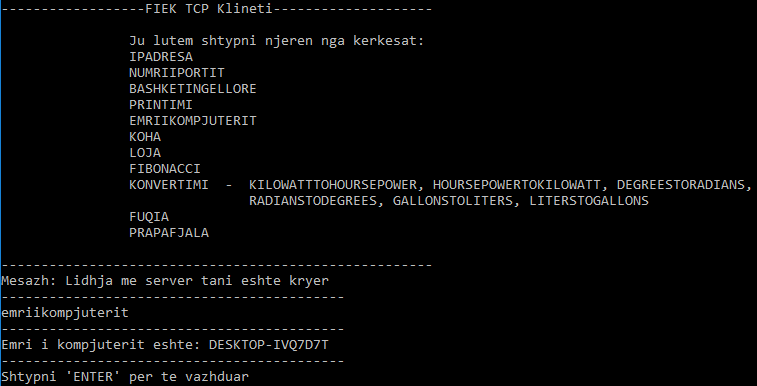


Figure5.Kthimi i emrit të kompjuterit me anë të metodës EMRIIKOMPJUTERIT

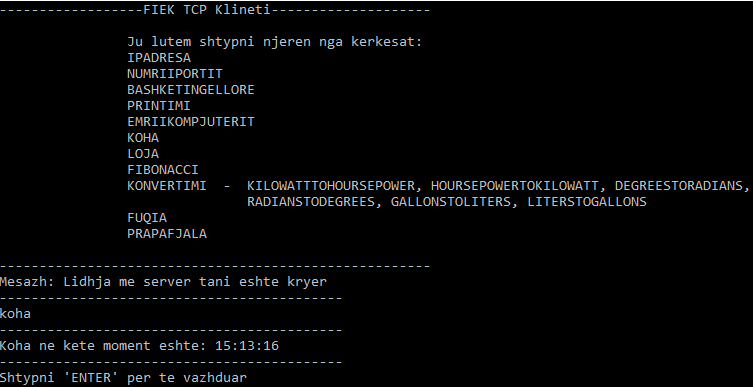
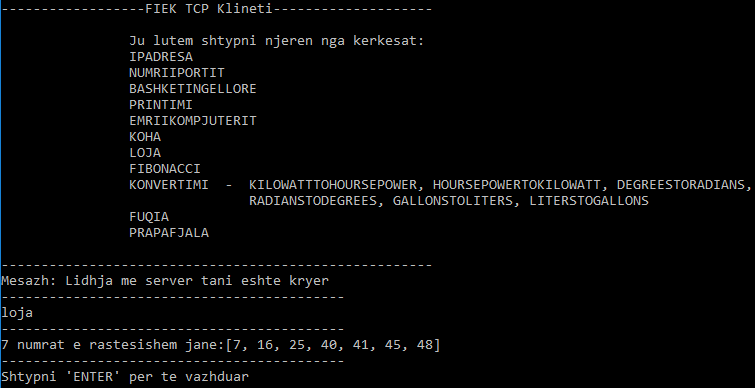


Figure 6.Kthimi i kohës me anë të metodës KOHA

  
Figure 7.Gjenerimi i numrave random me metodën LOJA

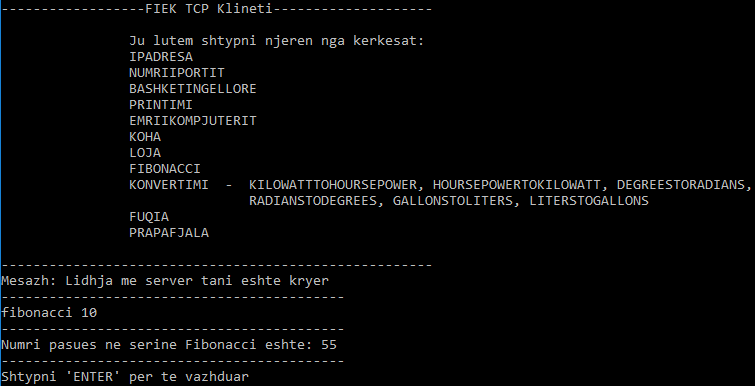


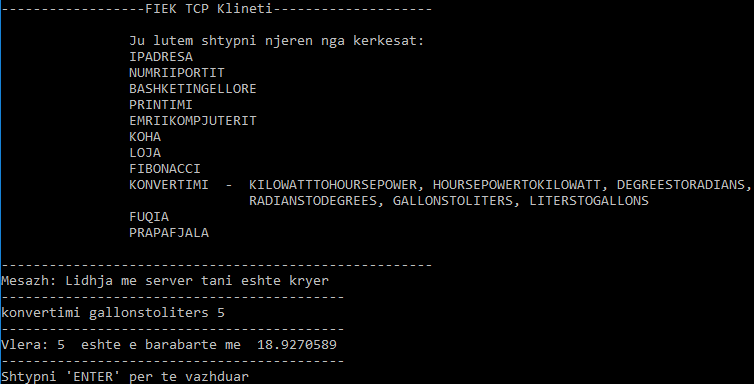
Figure 8.Gjetja e numrit pauses të sekuences Fibonacci m me metodën FIBONACCI

#### Metoda KONVERTIMI

Kjo metodë është më e nderlikuara e këtij projketi, pasi që në vete permban nën metodat e saja të cilat janë 6. Konvertimet e ndryshme prej:

* KilowattToHorsepower
* HorsepowerToKilowatt
* DegreesToRadians
* RadiansToDegrees
* GallonsToLiters
* LitersToGallons

Te cilat ia mundesojnë klientit që pas thirrjes së funksionit KONVERTIMI ,të thërrasë edhe nën metodat e këtij funksioni(të paraqitura në menynë e Klientit), e në bazë të zgjedhjes së Klientit programi do kthejë rezultatet e cilit do konvertim të dëshiruar.

 Figure 9.Konvertimi i një numri me metodën KONVERTIMI

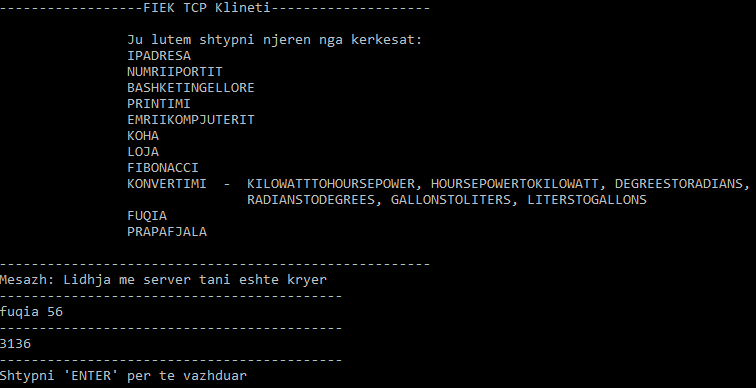
****

Figure 10.Llogaritja e fuqisë katrore përmes metodës FUQIA

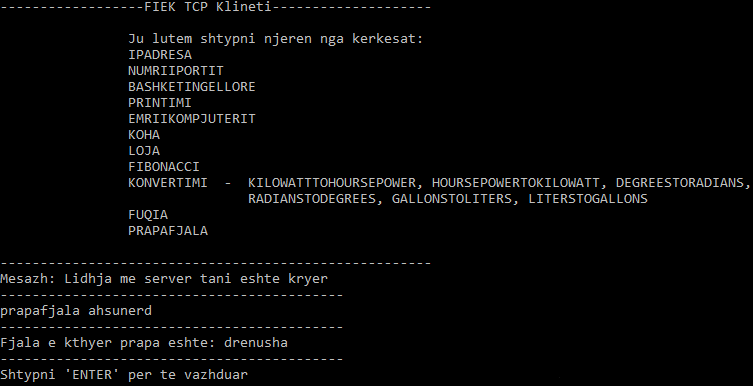


Figure 10.Kthimi I fjalës prapa përmes metodës PRAPAFJALA

#### Permbledhje rreth rezultateve

Sic mund të vërehet edhe nga print Screen-at e vendosur me lartë, secila metodë e dhënë funskionon, dhe secila prej tyre kthen rezultatin i cili realisht duhet të kthehet dhe është i sakti!

1. https://en.wikipedia.org/wiki/Transmission\_Control\_Protocol [↑](#footnote-ref-2)
2. https://en.wikipedia.org/wiki/User\_Datagram\_Protocol [↑](#footnote-ref-3)
3. https://en.wikipedia.org/wiki/Socket [↑](#footnote-ref-4)