

**FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE**

**DEPARTAMENTI I INXHINIERISË KOMPJUTERIKE**

PROJEKTI I PARË - FIEK PROTOCOL

**Lënda: Rrjeta Kompjuterike**

Punoi:

**Ida Spahiu**

**ID: 180714100125**

Mentorët:

**Prof. Dr. Blerim Rexha**

**Prof. Haxhi Lajqi**

**PERMBAJTJA**

HYRJA……………………………………………………………………………………

FIEK-TCP PROTOCOL………………………………………………………………….

* FIEK TCP CLIENT……………………………………………………..
* FIEK TCP SERVER…………………………………………………….

FIEK-UDP PROTOCOL………………………………………………………………….

* FIEK UDP CLIENT……………………………………………………..
* FIEK UDP SERVER……………………………………………………

PERSHKRIMI I METODAVE…………………………………………………………...

* IP ADDRESS……………………………………………………………
* PORT …………………………………………………………………....
* COUNT………………………………………………………………….
* TIME…………………………………………………………………….
* GAME…………………………………………………………………...
* PALINDROME………………………………………………………….
* REVERSE……………………………………………………………….
* GCF……………………………………………………………………...
* CONVERT………………………………………………………………

METODAT E ZGJEDHURA…………………………………………………………….

* NAME…………………………………………………………………..
* PRIME…………………………………………………………………..
* ENCRYPT………………………………………………………………
* DECRYPT………………………………………………………………
* NONE…………………………………………………………………...

KONKLUZIONET……………………………………………………………………….

REFERENCAT…………………………………………………………………………..

HYRJA

*Projekti i pare ne lëndën e rrjetave kompjuterike ka pasur për qëllim familjarizimin e studentëve me Socket Programming. Socket Programming apo i njohur edhe si Programimi me Soketa, është një mënyrë për t’i lidhur dy nyje në rrjet që ato të komunikojnë mes vete.*

*Socketa identifikohet nga IP adresa si dhe numri i portit, keto dy kompnente jane ato që e mundësojnë komunikimin ndërmejt socketave.*

*Njera nyje socket dëgjon në një port të vecantë tek një IP, ndërsa nyja tjetër socket mbërrinë tek tjetra në mënyrë që të krijojnë një lidhje, Serveri është ai i cili luan rolin e dëgjuesit ndërsa klienti provon të lidhet me të.*

Në kete projekt, detyra jonë ka qenë të krijojmë një TCP-protocol dhe një UDP-protocol të emëruar **FIEK**, **.** Për realizimin e këtij projekti është përdorur vegla **Visual Studio** dhe gjuha programuese **Python**, sistemi operativ **Windows Pro**, verzioni **1809.**

**FIEK-TCP PROTOCOL**

TCP – Transmission Control Protocol, është një nga protokollet me të rëndësishme. Ky protokoll përcakton mënyrën e krijimit dhe mirëmbajtjes së komunikimit në rrjet., përmes të cilit aplikacionet e ndryshme shkëmbejnë të dhëna. TCP punon me anë të IP-se, e cila definon se si kompjuteret shkëmbejnë paketa me të dhëna. Në kohët e sotme TCP dhe IP janë rregullat kryesore qe e përcaktojnë internetin. Programimi me socketa fillon atehere kur importohet libraria **socket** dhe krijohet objekti:



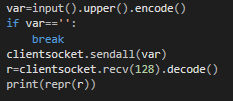
Këtu është bërë instanca e socketës, dhe futja e dy parametrave. **AF\_INET** i referohet familjes së adresave ipv4, ndërsa **SOCK\_STREAM** i referohet TCP protokollit.

* ***FIEK-TCP CLIENT***

Klienti është a i cili bën kërkesat, pra ky tenton të lidhet me serverin dhe në momentin që lidhet me serverin, ai bën kërkesën.Për tu lidhur me serverin duhet që të caktohet porti dhe IP adresa. Klienti ka mundësi që të zgjedhë vet portin dhe IP adresen me të cilin ai dëshiron të lidhet.

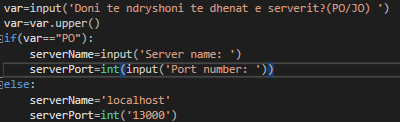
Për ta lidhur klientin me server:

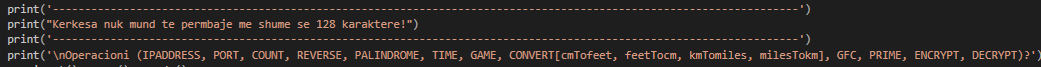


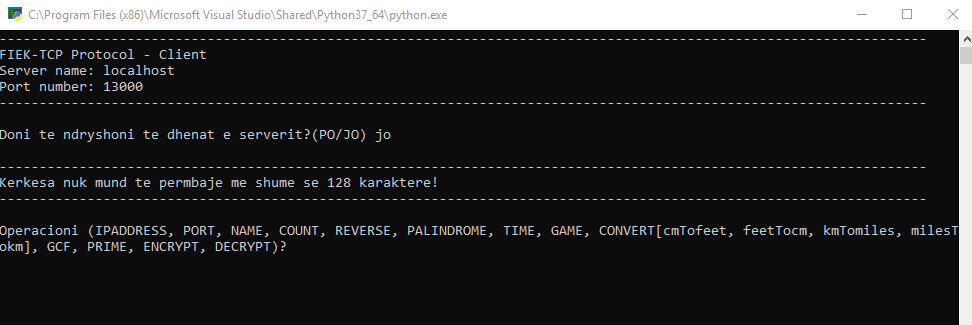


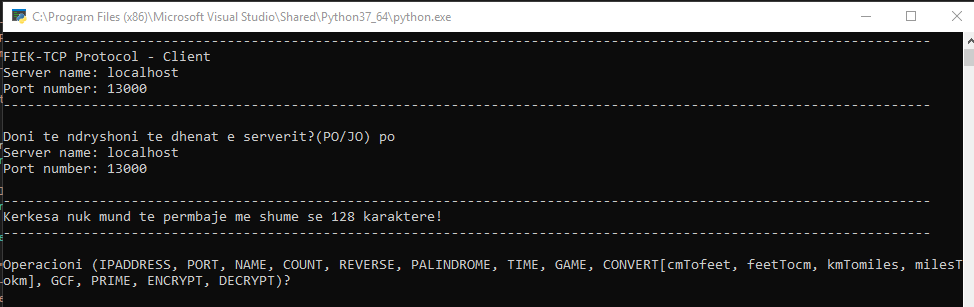
Disa nga metodat e shfrytëzuara nga kilenti për të komunikuar me serverin janë: **sendall(), recv(), encode(), decode(). Send()** dhe **recv()** shërbejnë për shkëmbimin e të dhënave ndërsa **encode()** dhe **decode()** shërbejnë për enkodimin dhe dekodimin e të dhënave.

Mundësia e përcaktimit të portit dhe IP adresës nga vetë klienti:



Poashtu, tek TCP Client mundësohet zgjedhja e njërës nga kërkesat e dhëna:

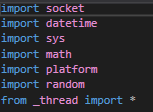
Ekzekutimi i FIEK-TC Protocol:

Ekzekutimi kur klienti i jep të dhënat e serverit:

* ***FIEK-TCP SERVER***

Programi server i dëgjon kërkesat të cilat vijnë nga klienti dhe i kthen përgjigje klientit. Serveri është në gjendje që të pranojë kërkesa nga disa klient. Kjo dukuri njihet si multithreading. Server i krijuar ka numrin e portit **13000**.

Tek programi TCP-SERVER perveq libraries socket janë importuar edhe library të tjera të cilat kanë qenë të nevojshme për krijimin e funksioneve.

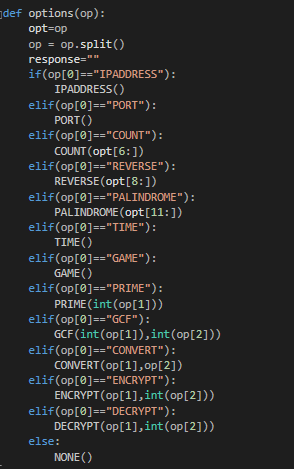


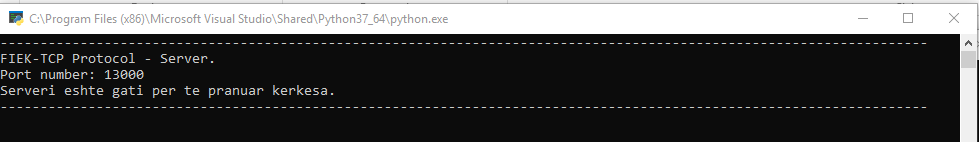
Disa nga metodat e perdorura në programin TCP-SERVER janë: **bind(), listen(),** **accept(), sendall(), recv(), encode(), decode().**

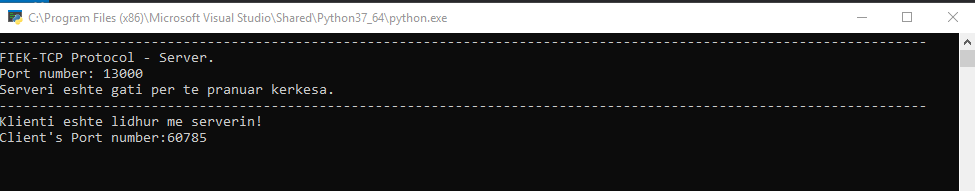


**Bind()** dhe listen janë ato që e karakterizojnë më shumë serverin. Bind mundëson që të lidhet në një port dhe IP adresë të caktuar, **listen()** i dëgjon kërkesat e klientit.

Në server jane implementuar të gjitha funksionet e kërkuara dhe është caktuar se cili funksion të thirret në rastin kur bëhet kërkesa nga klienti.



Ekzekutimi i TCP-SERVER Protocol:

Ekzekutimi kur serveri lidhet me klient:

**FIEK-UDP PROTOCOL**

UDP – User Datagram Protocol, është poashtu një protokoll i cili përcakton mënyrën e komunikimit mes dy pikave në rrjet. Për dallim nga TCP, ky i transmeton të dhënat mes dy pikave të ndryshme të cilat nuk kanë nevojë të lidhen paraprakisht. Nuk e bën kontrollimin e paketave të të dhënave të cilat shkëmbehen mes klientit dhe serverit, pra në rast të humbjes së paketave marrësit nuk i dërgohen përsëri.



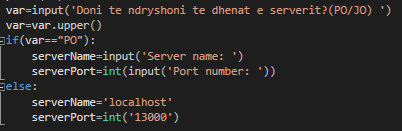
Këtu është bërë instanca e socketës, dhe futja e dy parametrave. **AF\_INET** i referohet familjes së adresave ipv4, ndërsa **SOCK\_STREAM** i referohet TCP protokollit

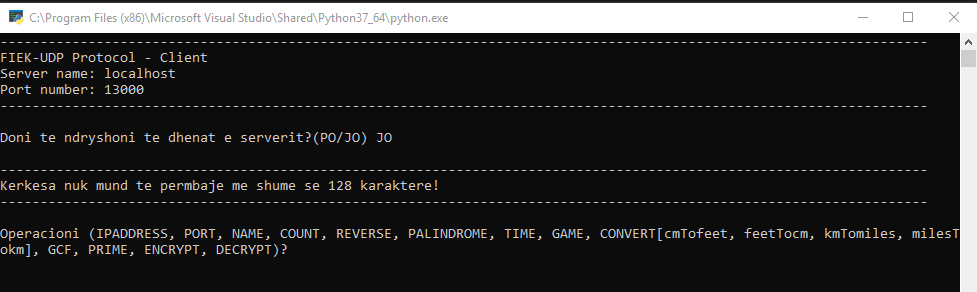
* ***FIEK-UDP CLIENT***

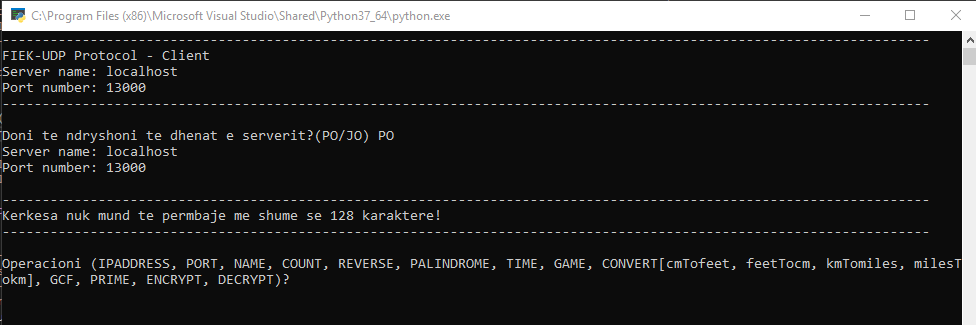
UDP klienti nuk formon lidhje me serverin, ai thjeshtë i dërgon atij një datagram.



Edhe tek UDP klienti, klientit i ofrohet mundësia për të zgjedhur vetë portin dhe IP adresën.



Ekzekutimi i FIEK-UDP Client:

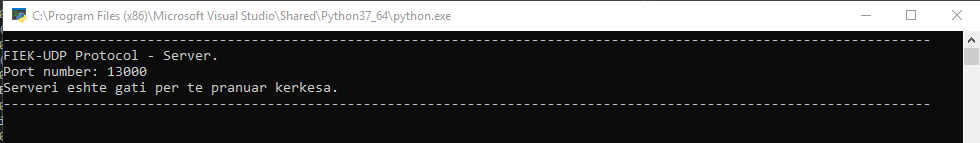
Ekzektuimi kur të dhënat jepen nga klienti:

* ***FIEK-UDP SEVER***

Serveri tek UDP protokolli nuk pret që të krijohet lidhja, thjeshtë pret që të arrijë datagrami. Datagram i cili arrin tek klienti, përmban edhe adresën e dërguesit dhe kjo ia mundëson serverit që të dijë se ku ta dërgojë përgjigjen saktësisht.



Edhe tek UDP-SERVER njejte sit e TCP funksionet janë implementuar në server si dhe është caktuar se cili funksion të thirret varësisht nga kërkesa e klientit.

Ekzekutimi i FIEK-UDP Server:

**PËRSHKRIMI I METODAVE**

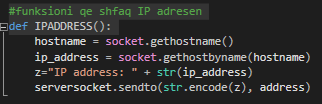
Metodat të cilat janë dhënë si kërkesë në projekt janë:

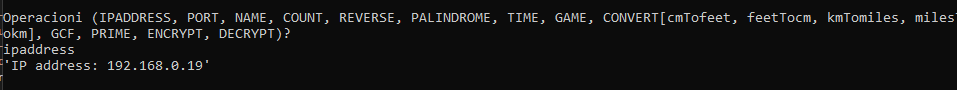
1. IPADDRESS()
2. PORT()
3. COUNT()
4. TIME()
5. GAME()
6. PALINDROME()
7. REVERSE()
8. GCF()
9. CONVERT()

Në vijim do të gjeni përshkrimin e secilës metodë, mënyrën e krijimit si dhe nga një shembull për secilën. Të gjitha metodat ekzekutohen ashtu sic është kërkuar!

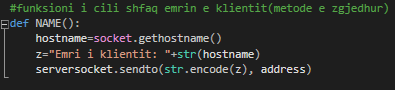
* ***IPADDRESS()***

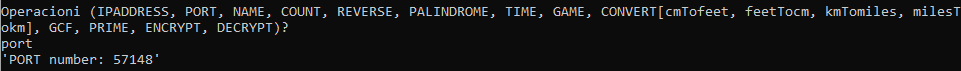
Kjo metodë përdoret për të kthyer IP adresën e klientit.





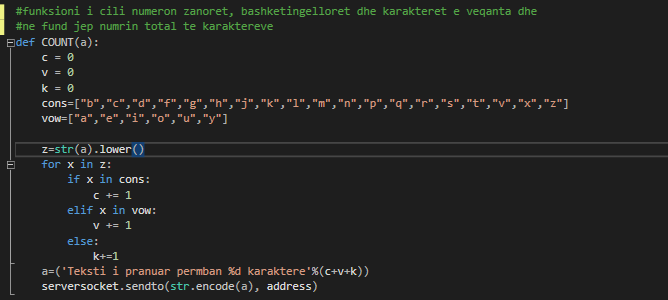
* ***PORT()***

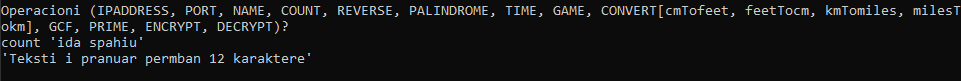
Kjo metodë përdoret për të kthyer portin e klientit.



* ***COUNT()***

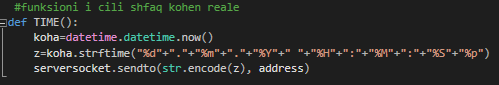
Metoda count shërben për të numëruar zanoret, bashkëtingëllorer dhe karakteret speciale të një teksti dhe në fund kthen numrin e të gjitha karaktereve që përmban ai tekst.

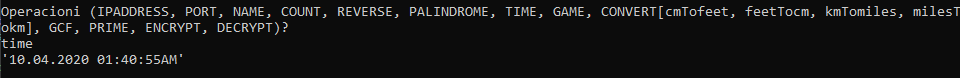




* ***TIME()***

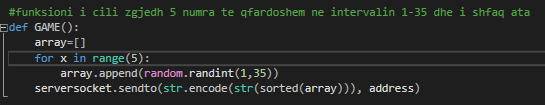
Kjo metodë shërben për të dhënë kohën e saktë në të cilën ndodhemi.

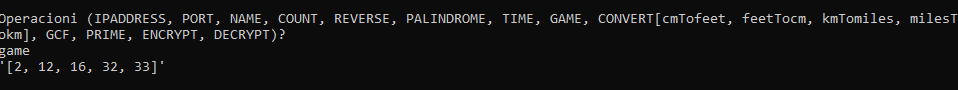




* ***GAME()***

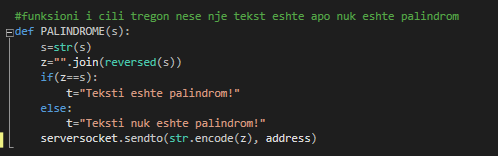
Kjo metodë shërben për të shfaqur 5 numra të cfarëdoshëm në intervalin [1, 35], numrat shfaqen te sortuar.

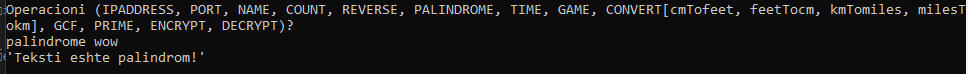




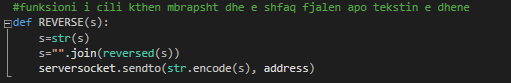
* ***PALINDROME()***

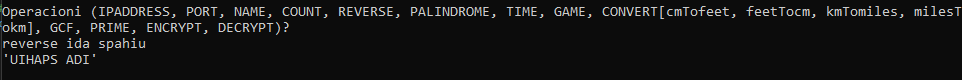
Metoda palindrome() tregon nëse teksti i dhënë është apo nuk është palindrom.



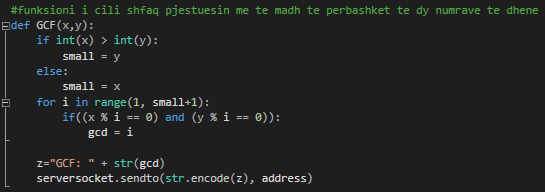


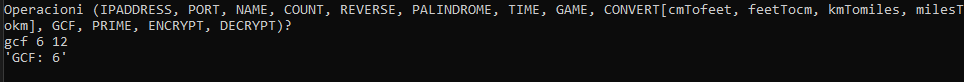
* ***REVERSE()***

 Metoda reverse() shërben për të kthyer mbrapsht një tekst të dhënë.



* ***GCF()***

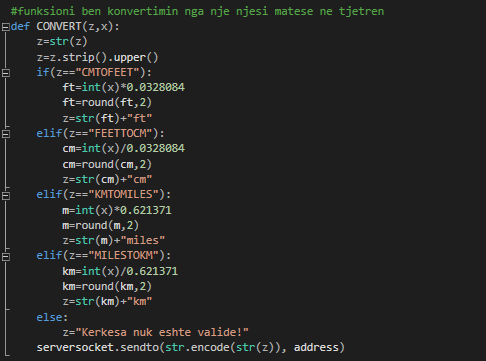
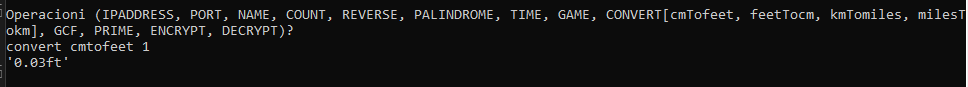
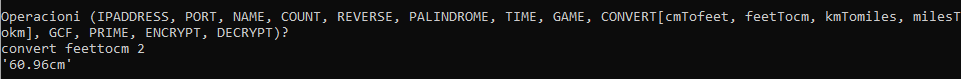
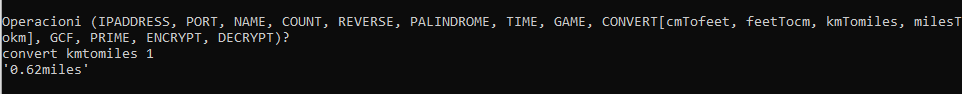
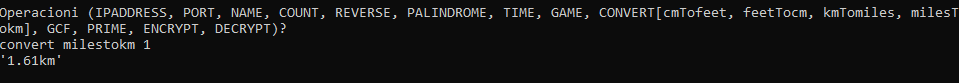
 Metoda gfc() kthen pjestuesin më të madh të përbashkët të dy numrave të dhënë.



* ***CONVERT()***

Metoda convert() i ofron klientit të bëjë konvertimin e njësive matëse:

1. cmTofeet
2. feetTocm
3. kmTomiles
4. milestokm

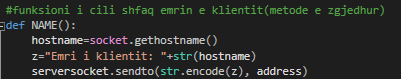
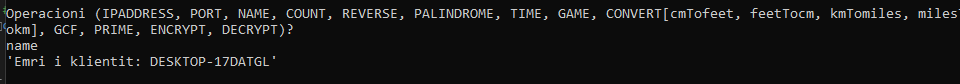
Argument i parë është lloji i konvertimit, i dyti është vlera që do të konvertohet.

**METODAT E ZGJEDHURA**

Metodat qe kam përzgjedhur jane:

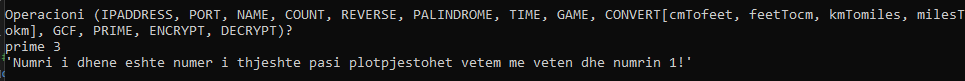
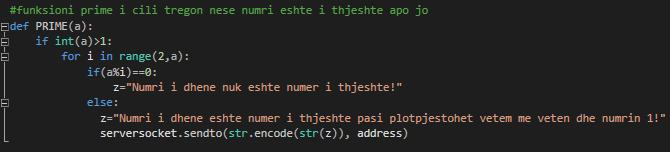
1. NAME()
2. PRIME()
3. ENCRYPT()
4. DECRYPT()
5. NONE()

* ***NAME()***

Metoda name() si rezultat shfaq emrin e klientit.

* ***PRIME()***

Metoda prime() tregon nëse një numër është i thjeshtë apo jo. Për kërkesën prime duhet të shkruhet : ***prime***[hapsirë]***vlera***

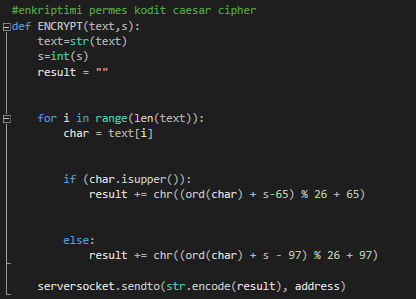


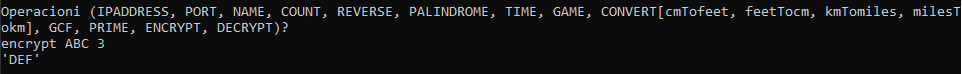
* ***ENCRYPT()***

Metoda encrypt() bën enkriptimin e tekstit me anë të Caesar cipher komandës.

Për kërkesën encrypt duhet të shkruhet: ***encrypt***[hapsirë]***teksti***[hapsirë]***celësi.***

Celësi është numër, tregon se për cfarë vlere do të zhvendosen shkronjat e tekstit!



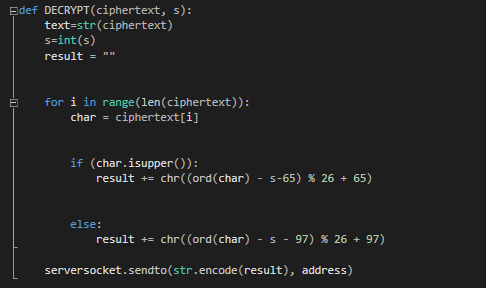
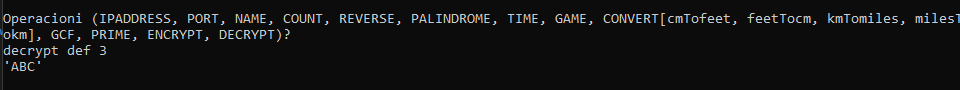


* ***DECRYPT()***

Metoda decrypt() bën dekriptimin e tekstit me anë të Caesar cipher komandës.

Për kërkesën decrypt duhet të shkruhet: ***decrypt***[hapsirë]***teksti***[hapsirë]***celësi.***

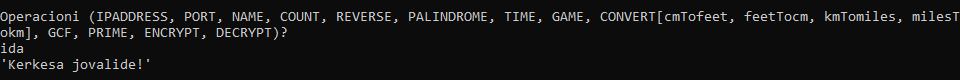
Celësi është numër, tregon se për cfarë vlere do të zhvendosen shkronjat e tekstit!



* ***NONE()***

Metoda none() kthen tekstin “Kërkesa jovalide!”, nëse klienti shkruan diqka që nuk ka lidhje me metodat e mësipërme.





**KONKLUZIONI**

Duke krahasuar kërkesat e projektit dhe ekzekutimin e programeve, mendoj që kërkesat janë plotësuar ashtu sic është kërkuar. Të gjitha komandat ekzekutohen dhe kthejnë rezultat, në rastin kur klienti jep ndonjë kërkesë që nuk ekziston, programi e merr kërkesën si jovalide dhe vazhdon ekzekutimin. Poashtu është përdorur metoda multithreading e cila lejon qasjen e më shumë se një klienti në server. Janë implementuar gjithsej 4 metoda shtesë, dhe është bërë përshkrimi se si funksionojnë ato metoda. Metodat të cilat na janë caktuar ekzekutohen ashtu si është demonstruar në projekt. Gjitashtu janë përfshirë edhe error handling, të cilët mundësojnë stabilizimin e programit në rast të ndonjë gabimi.

Përndryshe, programimi me socketa ishte një eksperiencë mjaft e këndshme dhe argëtuese !

**REFERENCAT**

<https://realpython.com/python-sockets/>

<https://www.geeksforgeeks.org/python-programming-language/>

<https://www.w3resource.com/>

<https://www.w3schools.com/python/>

**JU FALEMINDERIT!**