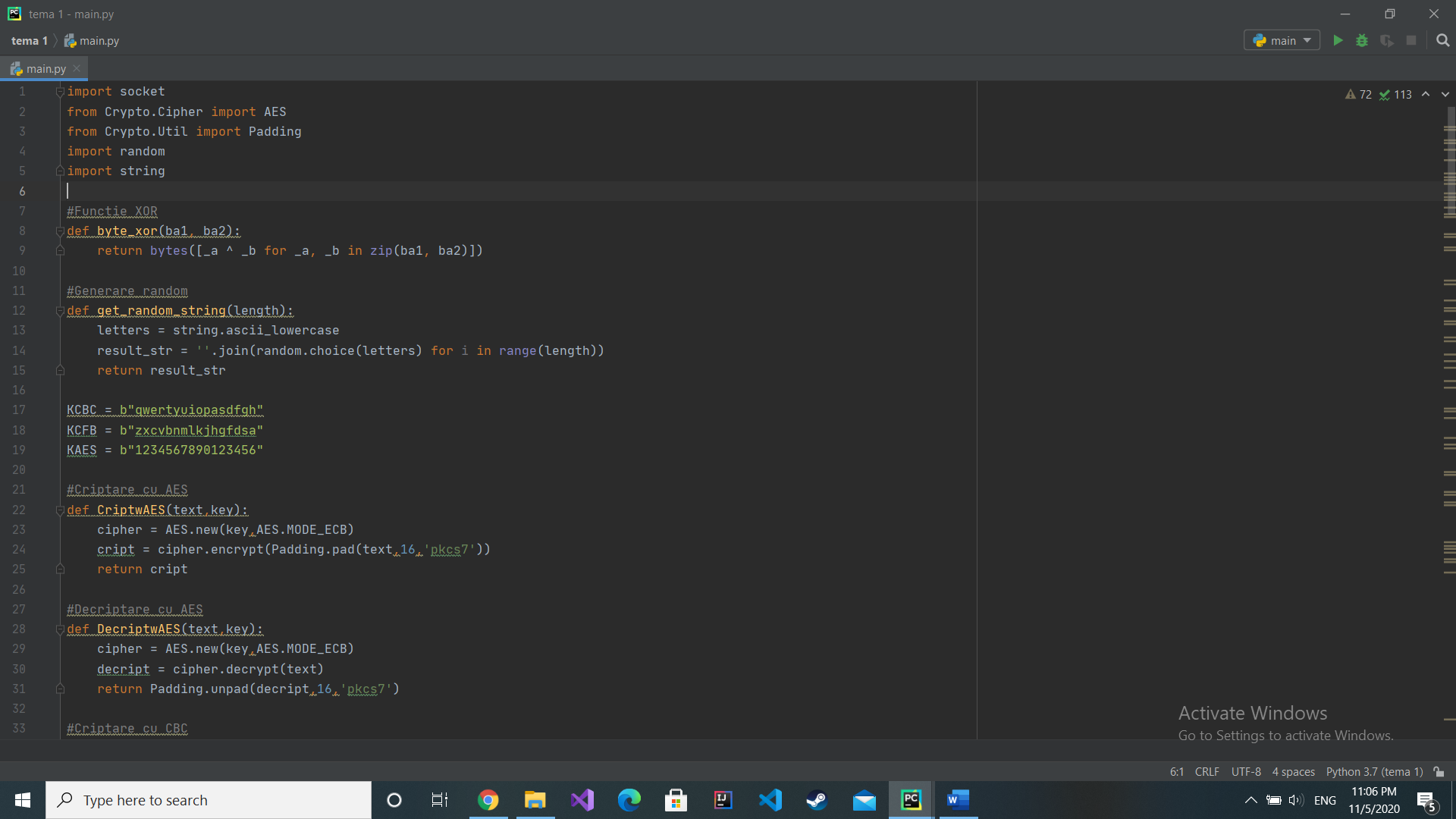
**Documentatie Tema1**

**Securitatea Informatiei**

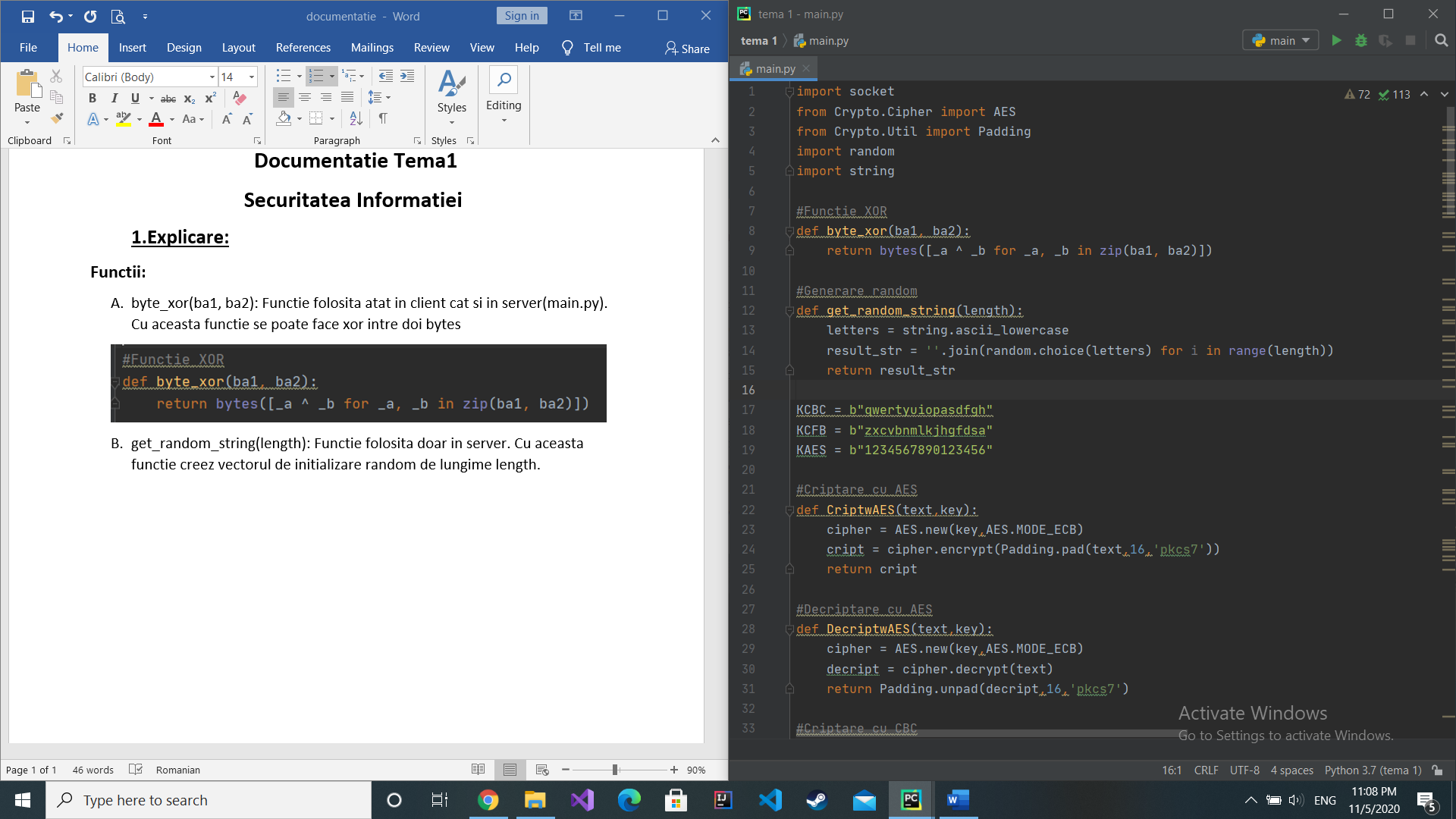
**1.Explicare:**

**Functii:**

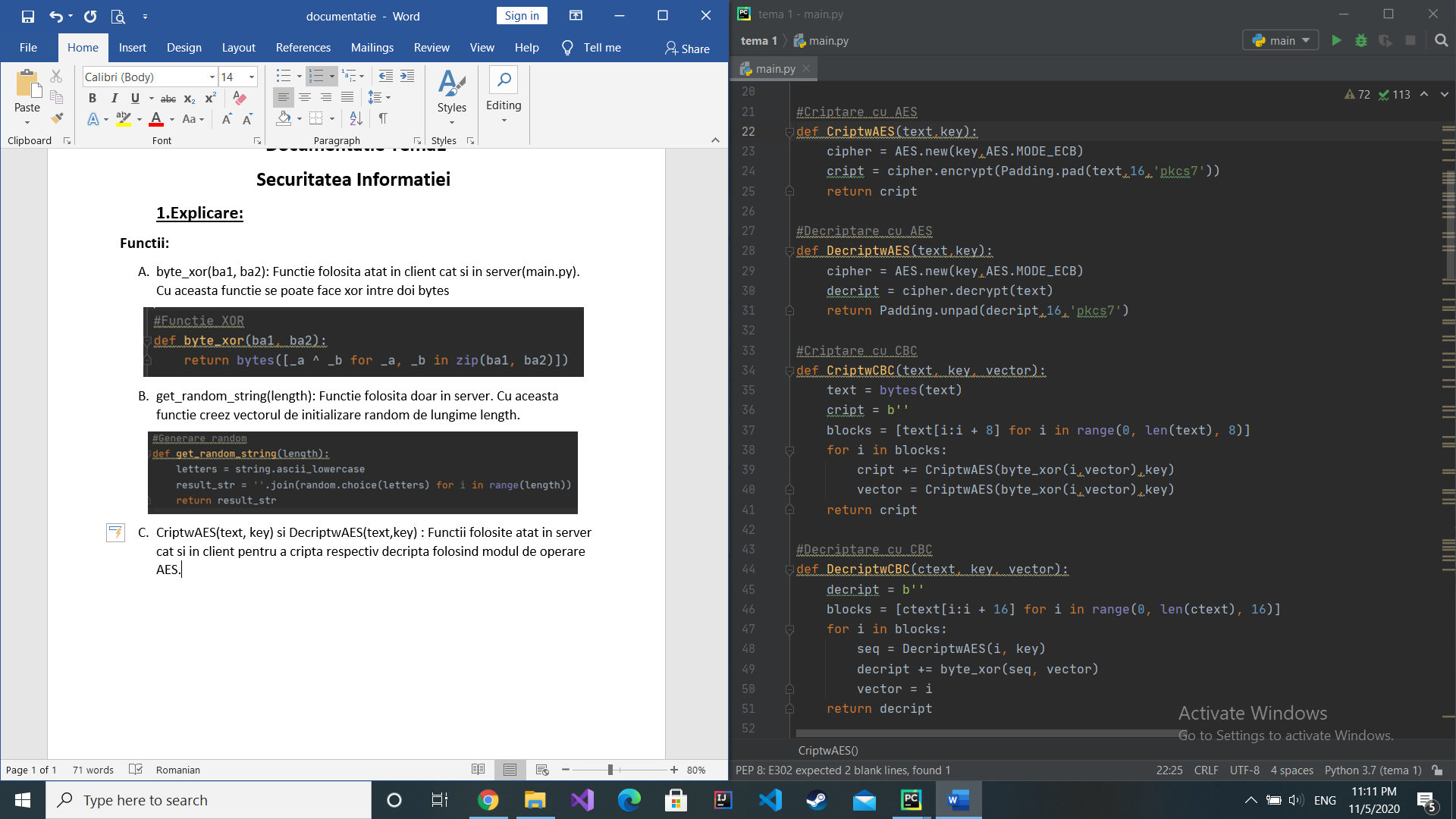
1. byte\_xor(ba1, ba2): Functie folosita atat in client cat si in server(main.py). Cu aceasta functie se poate face xor intre doi bytes



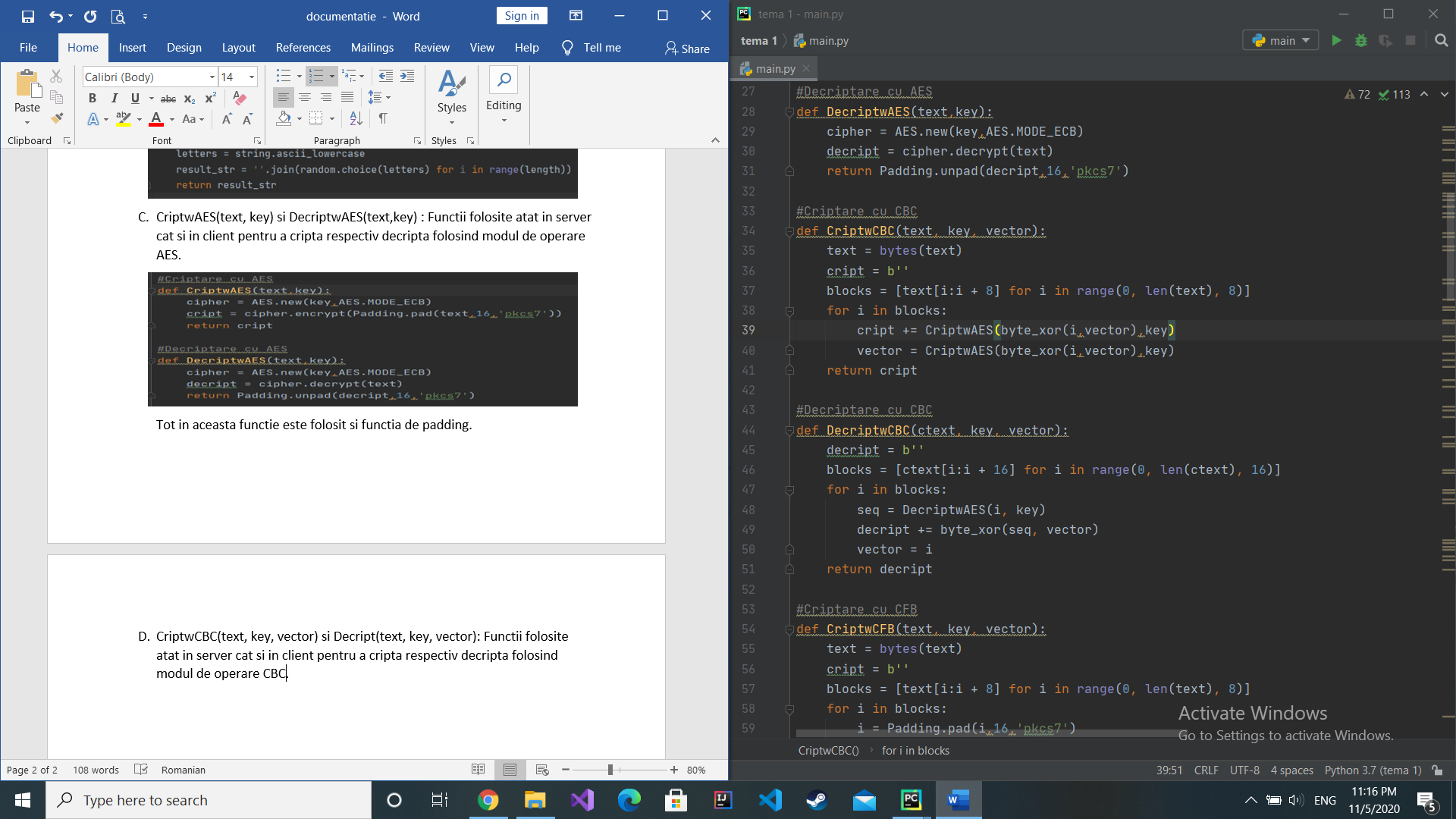
1. get\_random\_string(length): Functie folosita doar in server. Cu aceasta functie creez vectorul de initializare random de lungime length.



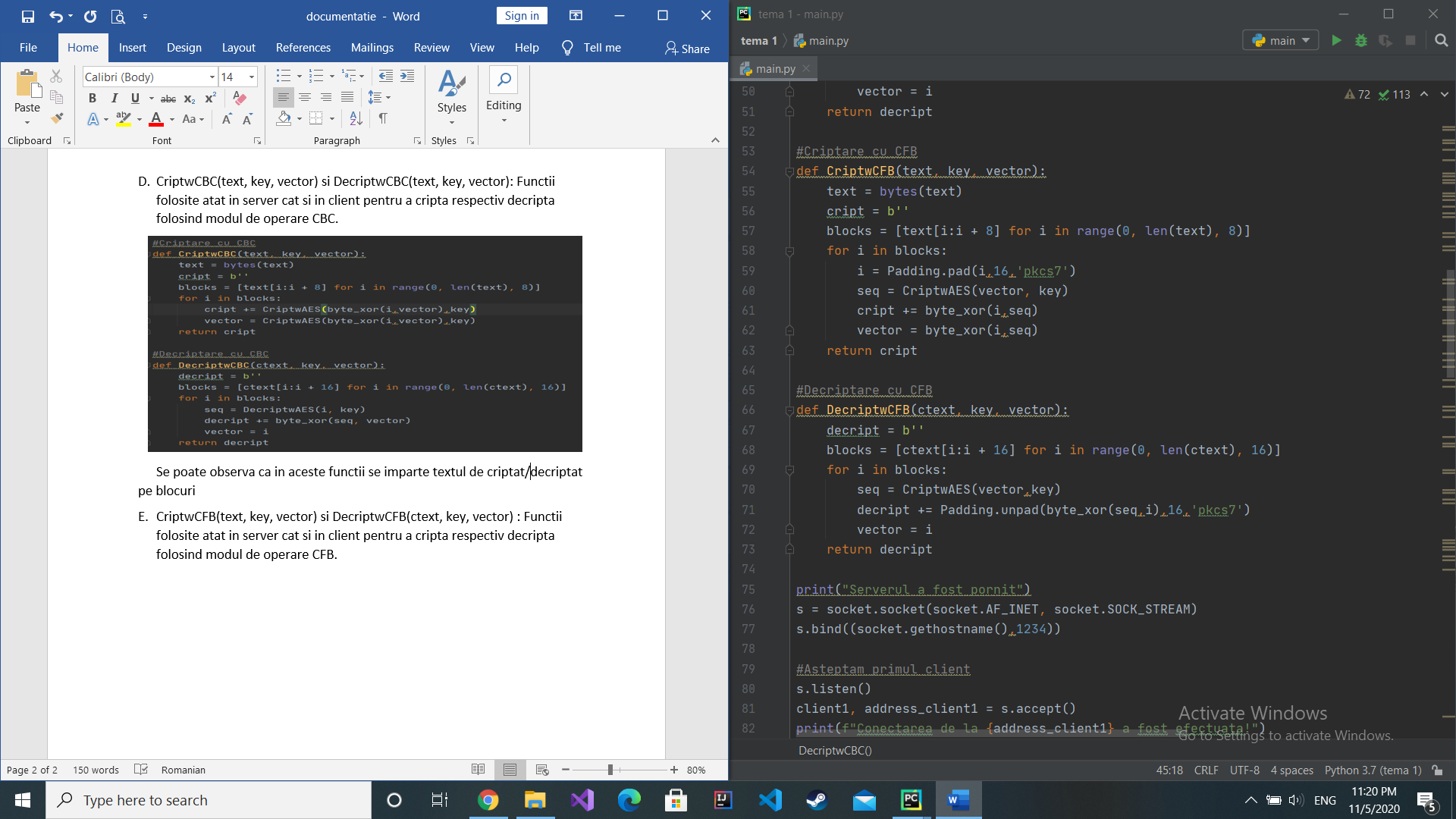
1. CriptwAES(text, key) si DecriptwAES(text,key) : Functii folosite atat in server cat si in client pentru a cripta respectiv decripta folosind modul de operare AES.

Tot in aceasta functie este folosit si functia de padding.

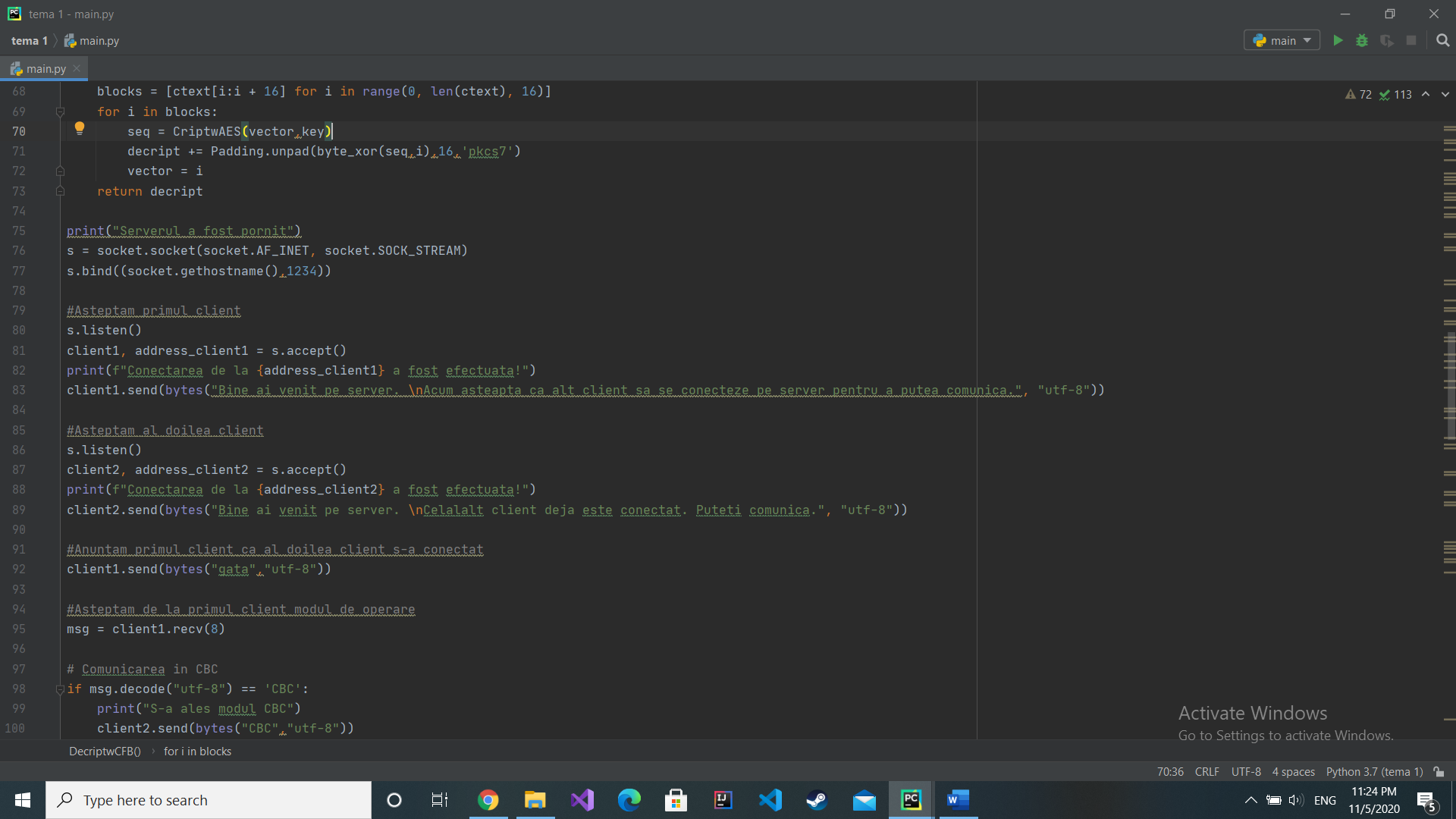
1. CriptwCBC(text, key, vector) si DecriptwCBC(text, key, vector): Functii folosite atat in server cat si in client pentru a cripta respectiv decripta folosind modul de operare CBC.

Se poate observa ca in aceste functii se imparte textul de criptat/decriptat pe blocuri

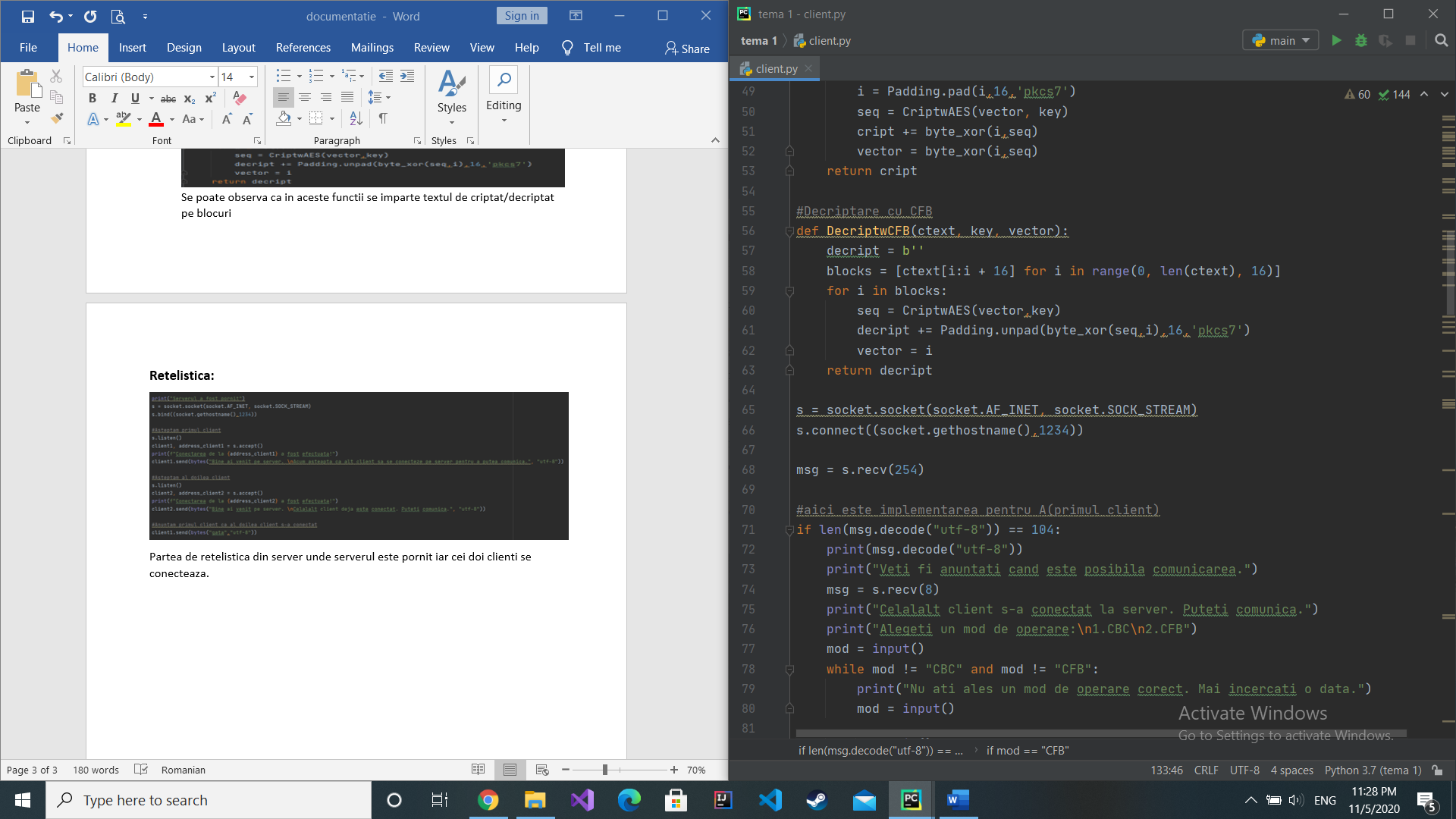
1. CriptwCFB(text, key, vector) si DecriptwCFB(ctext, key, vector) : Functii folosite atat in server cat si in client pentru a cripta respectiv decripta folosind modul de operare CFB.

Se poate observa ca in aceste functii se imparte textul de criptat/decriptat pe blocuri

**Retelistica:**



Partea de retelistica din server unde serverul este pornit iar cei doi clienti se conecteaza.



Partea de retelistica din client unde clientii se conecteaza la server.

**Restul programului:**

Restul programului reprezinta exact comunicarea intre cei doi clienti folosindu-se de server.

In client.py codul este impartit in doua secvente: prima secventa reprezinta Clientul A, iar a doua Clientul B, Clientul 1 fiind construit intre liniile 70 – 157 iar Clientul B intre liniile 158 – 237

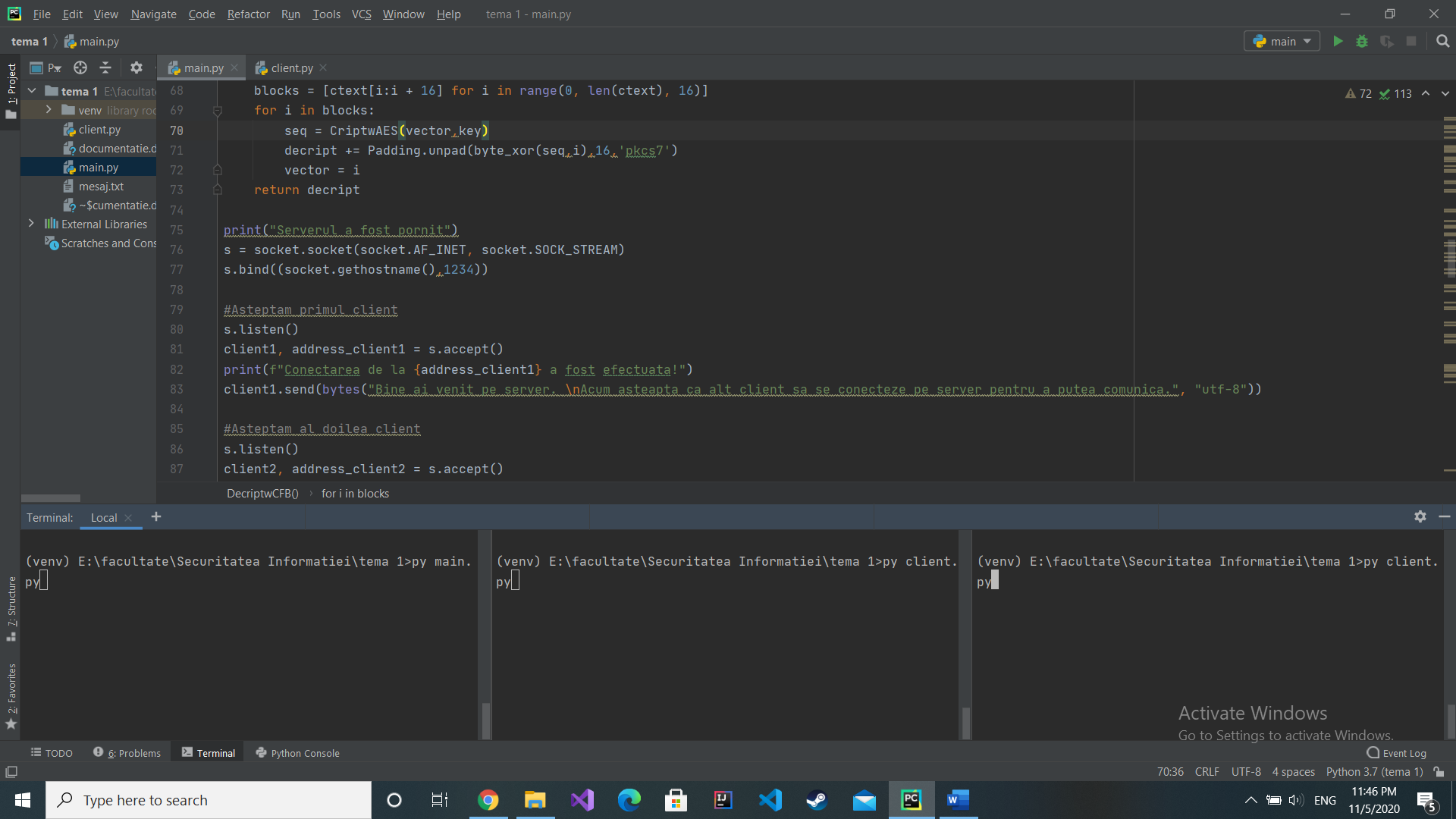
Fiecare client este impartit in cate doua secventa, prima secventa reprezentand comunicarea folosind modul de operare CBC iar a doua secventa CFB.

**2.Atentie, exista un bug:**

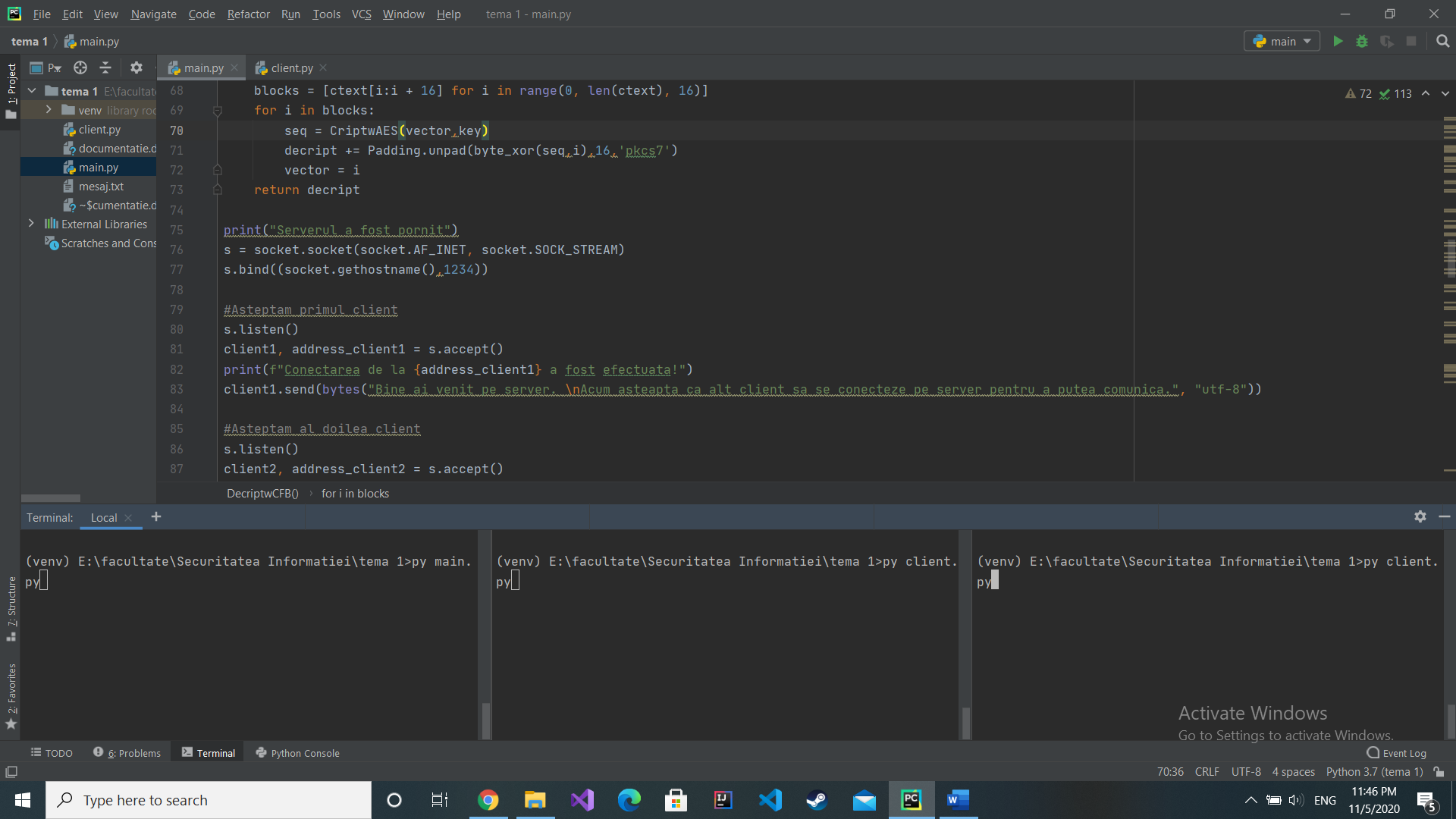
Cand clientul A alege modul de operare, daca alege modul intr-un timp prea rapid, al doilea client nu va mai functiona.

**3.Mod de rulare:**

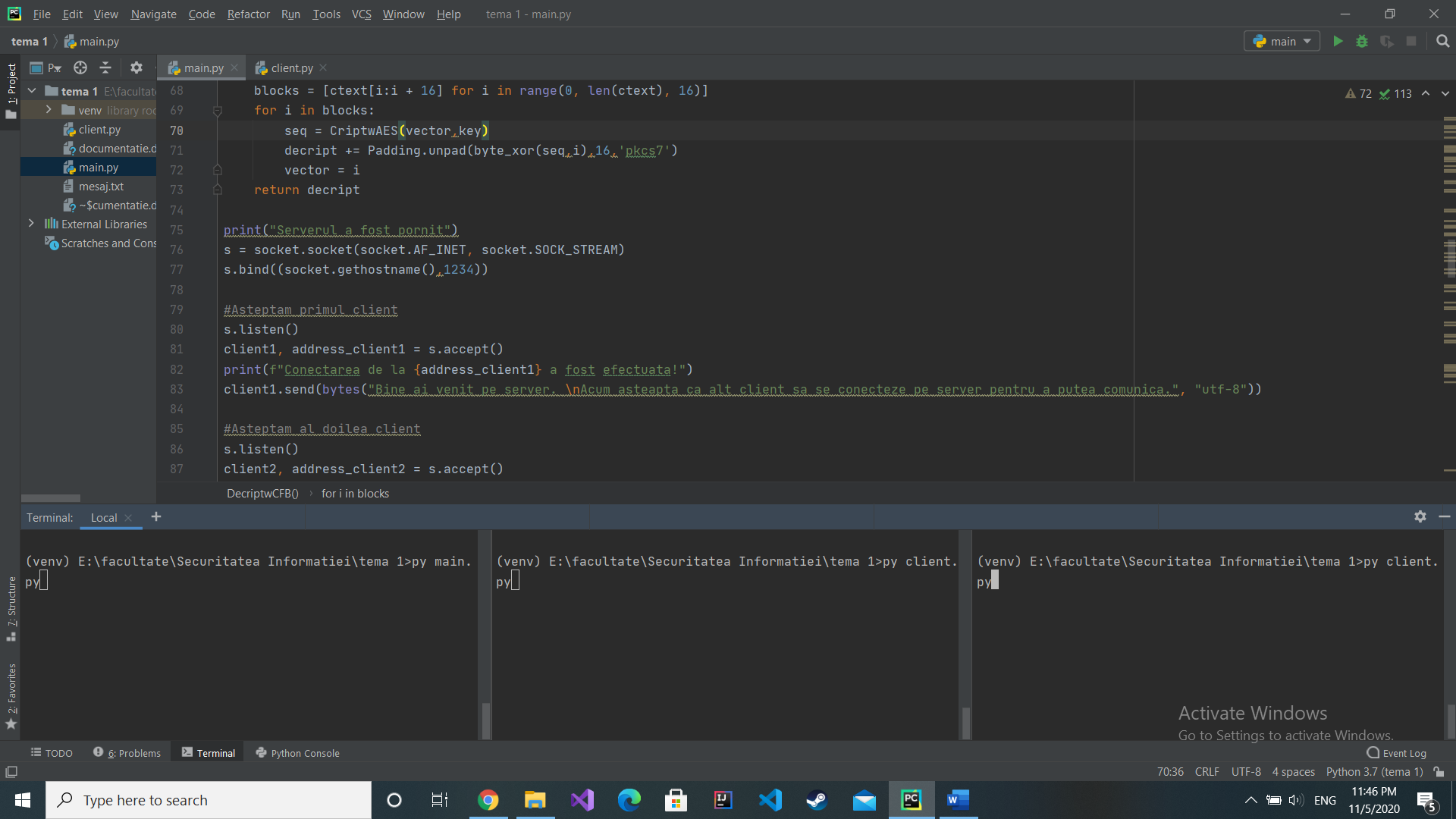
Se vor folosi 3 terminale: primul pentru server, al doilea pentru clientul A si al treilea pentru cientul B, serverul putand fi rulat cu comanda py main.py iar ambii clienti cu comanda py client.py.



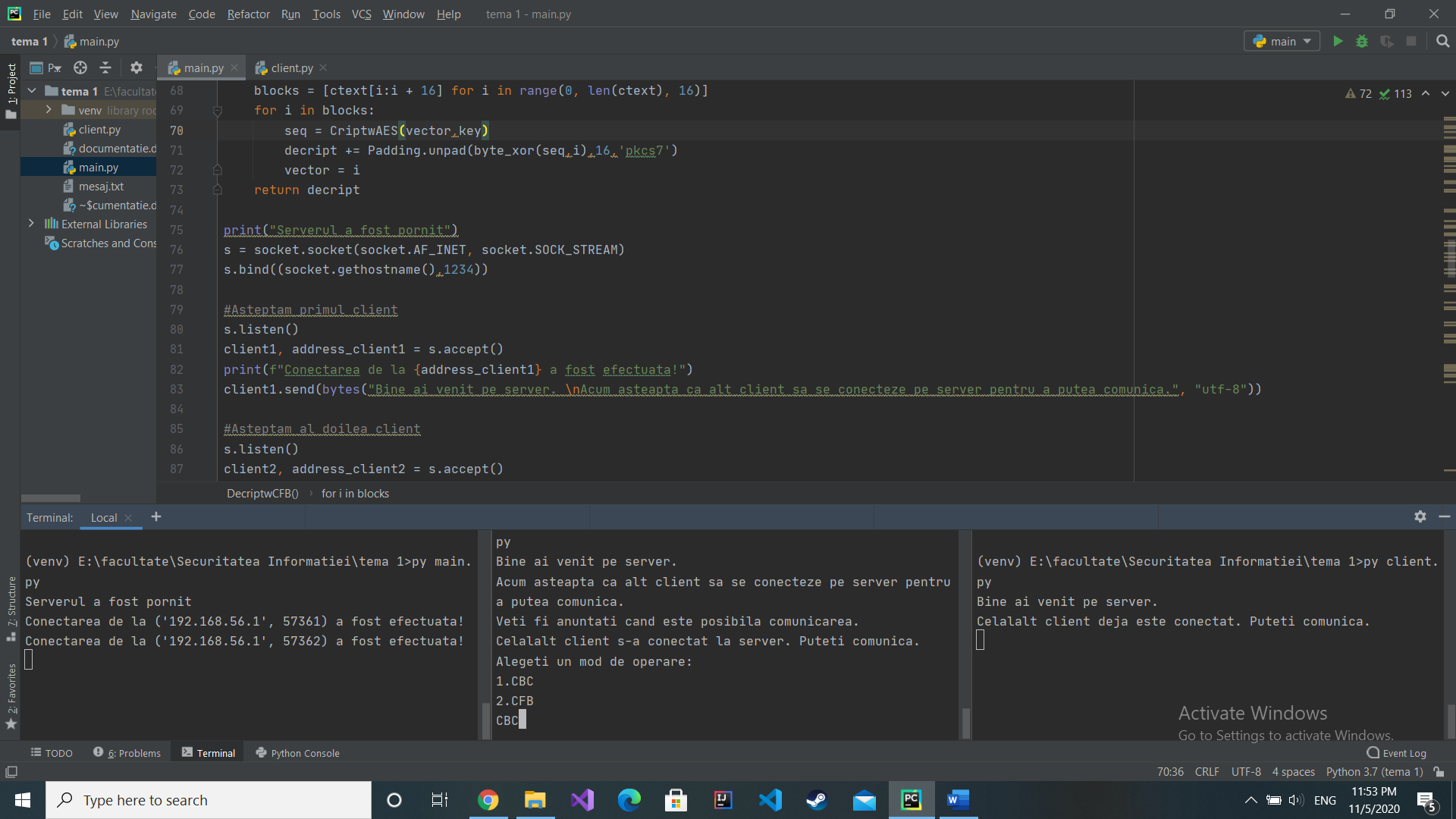
1. Prima data se va rula serverul.



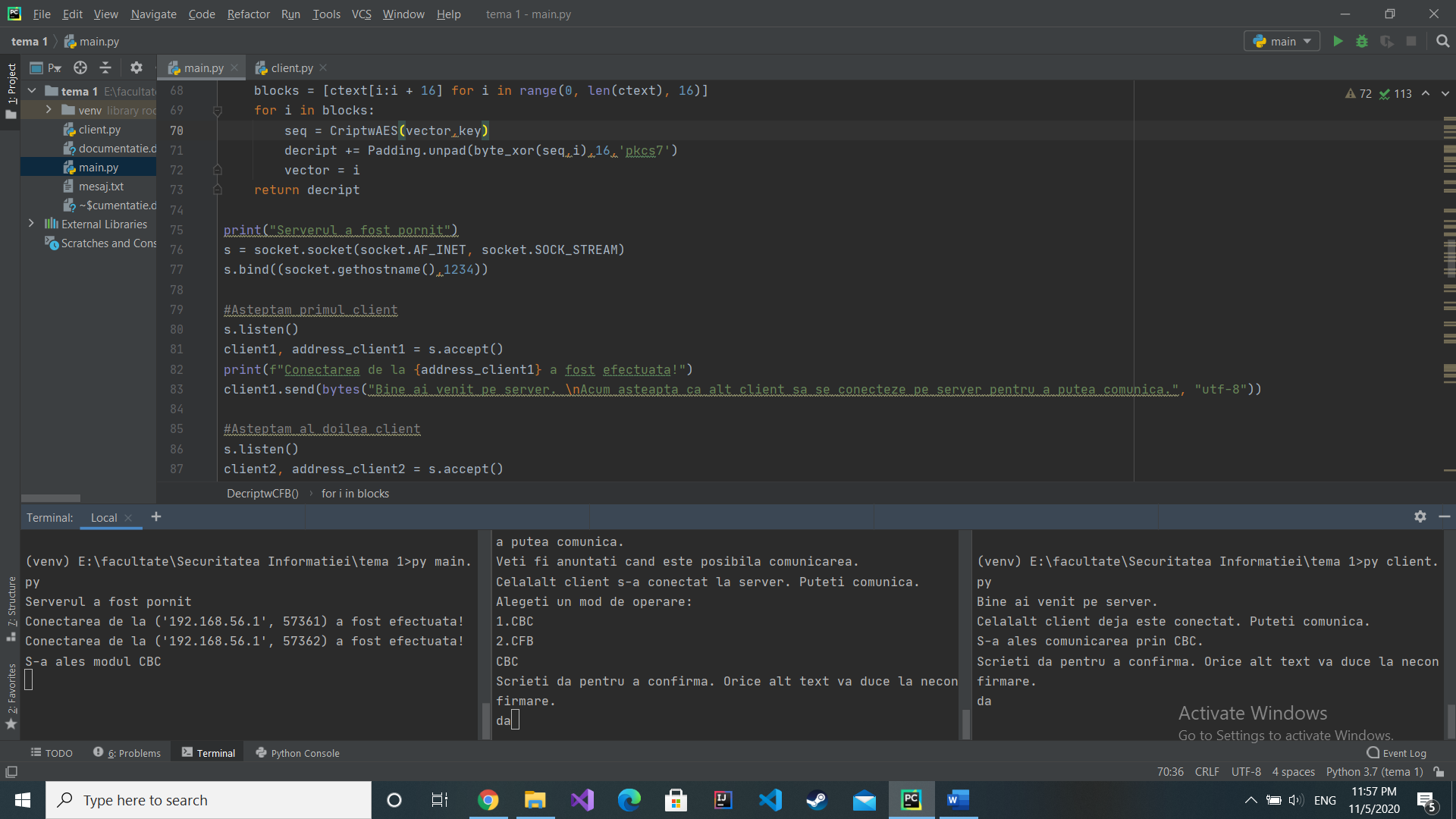
1. Dupa se va rula clientul, primind rolul de Clientul A.



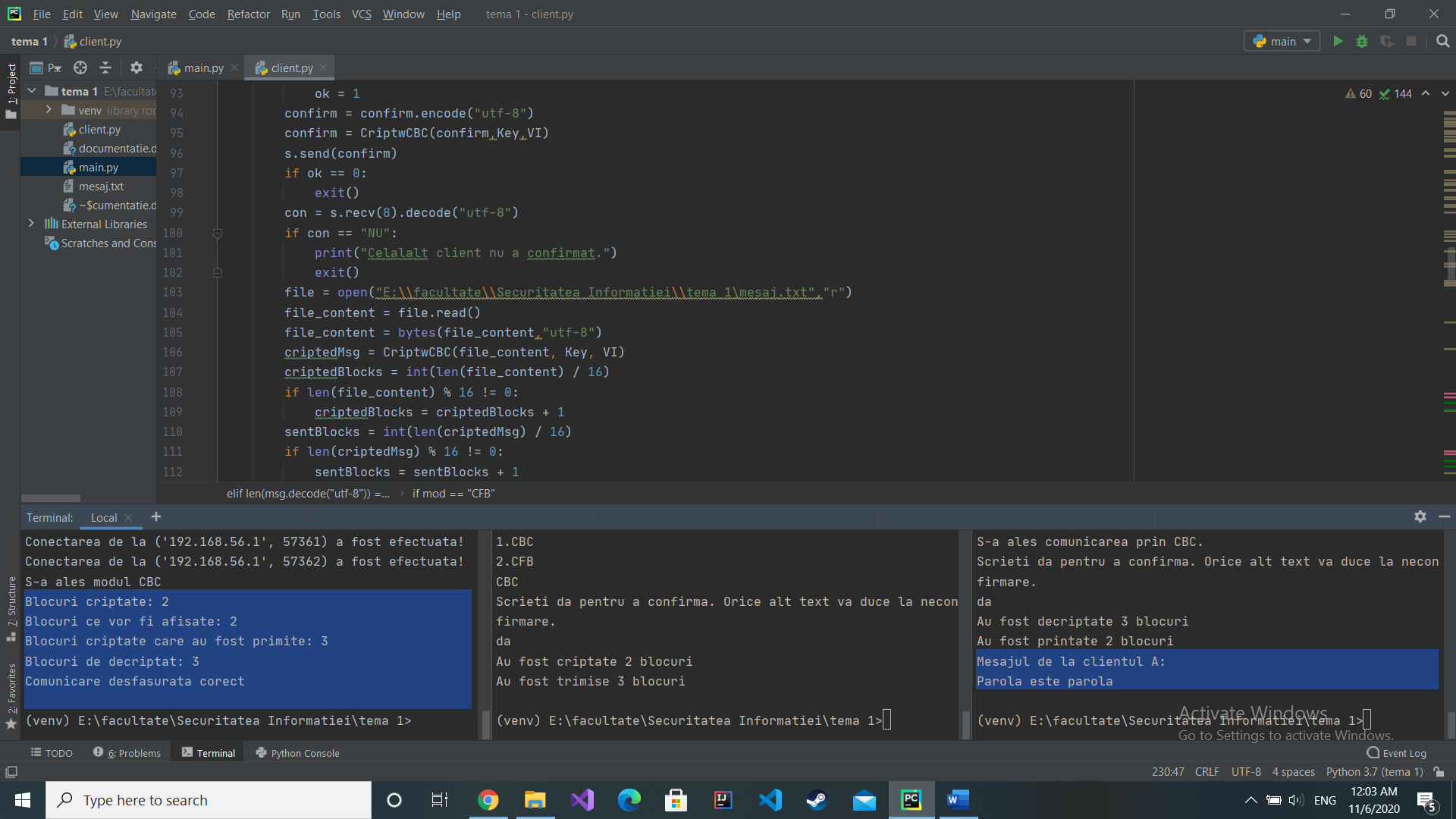
1. Dupa se va rula clientul din nou, primind rolul de Clientul B.



1. Dupa va trebui sa se aleaga modul de operare de catre Clientul A (CBC sau CBF). In cazul in care se scrie altceva programul te obliga sa reincerci.

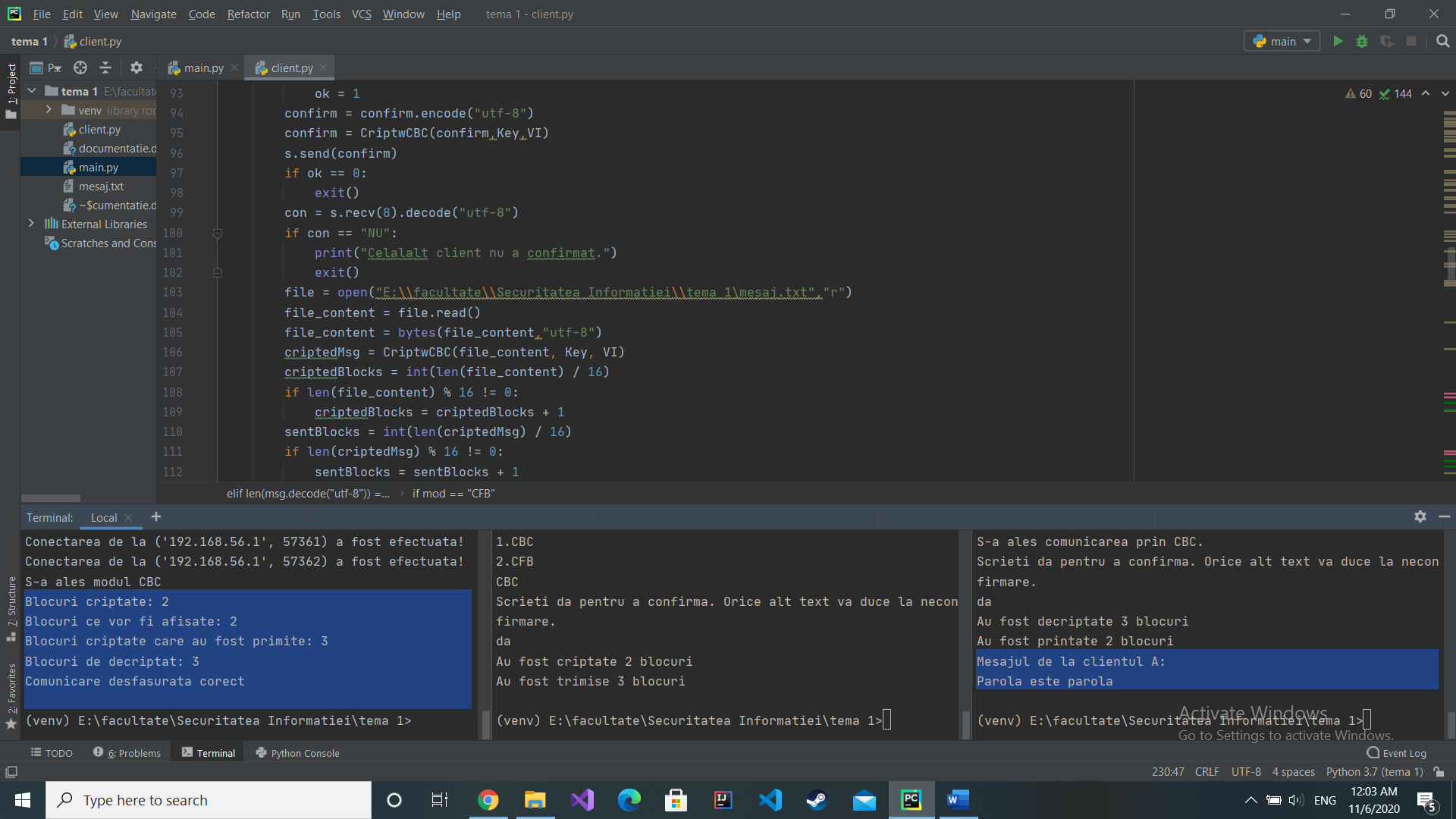


1. Ambii clienti trebuie confirme faptul ca doresc sa comunice scriind da. In cazul in care se scrie orice altceva atat serverul cat si clientii se vor inchide deoarece se va intelege ca unul din clienti nu au dorit sa confirme.



1. Dupa confirmarea celor doi cienti mesajul din Clientul A va ajunge la Clientul B criptat, decriptarea fiind efectuata in Clientul B.

OBS:Mesajul este preluat din fisier.



Serverul anunta daca comunicarea a fost efectuata intr-un mod corect comparand blocurile criptate, decriptate, primite si trimise.

