# Exercitiul 3

1. Pentru rezolvarea am folosit limbajul de programare Python
2. “cryptography” este librarie criptografica pe care am folosit-o ( <https://cryptography.io/en/latest/>)
3. Din cadrul librariei am folosit algoritmul de criptare (EBC) si functia de padare
4. Problema a fost rezolvata prin intermediul unei aplicatii client/server
5. Pentru rularea exercitiul se vor deschide 3 terminale in care se vor scrie urmatoare , in ordinea in care urmeaza:
   1. python3 key\_manager\_server.py
   2. python3 clientA.py "CFB" "fisier.txt"
   3. python3 clientB.py
6. key\_manager.py
   1. Va astepta la portul 7123 conexiuni din partea clientilor ce doresc sa initieze o comunicare intre ei
   2. Clasa “Key Manager” indeplineste rolul de Key Manager , in functie de modul de operare cerut , acesta va oferi cheia de criptare criptata
7. clientA.py
   1. Se va conecta la “Key Manager” si va cere cheia de criptare si o decripta.
   2. Ii trimite lui B modul de operare si va astepta confirmarea acestuia
   3. Dupa ce primeste confirmarea acestuia trimite fiecare bloc de octeti intr-un mod criptat (din fisier sau citit blocuri de 128 de biti)
   4. Daca ultimul bloc nu are lungimea de 16 octeti acesta va fi padat pana va ajunge la acesta dimensiune
   5. Lui B i se trimite numarul de blocuri ce urmeaza sa fie trimise
8. clientB.py
   1. Asteapta de la A modul de operare
   2. Se conecteaza la Key Manager si ii cere cheia pentru modul de operare ales de A
   3. Citeste numarul de blocuri de vor fi trimise
   4. Asteapta fiecare bloc si il decripteaza si afiseaza mesajul decriptat
   5. Daca este cazul , face inversul operatiei de padare (Unpadding)
9. Toate mesajele transime au fost decriptate cu succes