# Exercitiul 2

1. Problema a fost rezolvata in limbajul c++
2. Pentru operatiile de criptare si decriptare am utilizat API-ul specificat in cerinta problemei si anume API EVP
3. In directorul Ex2 sunt urmatorele fisiere:

* Directorul **tests** care contine la randul sau 10 directoare in fiecare din care sunt doua fisiere file1.in si file2.in care difera doar printr-un octet
* **main.cpp** care reprezinta rezolvarea problemei propuse si anume
  + functie pentru generarea hash-ului SHA256(**getSHA256**) pentru continutul unui fisier si scrierea acestuia in fisierul specificat ca parametru
  + functie pentru generarea hash-ului MD5(**getMD5**) pentru continutul unui fisie si scrierea acestuia in fisierul specificat ca parametru
  + functie pentru compararea a doua fisere ce contit hash-uri in format hexa
  + si functia carea apeleaza pe acele enumerate mai sus **hash\_and\_compare**
    - aceasta genereza hash-urile SHA256 si MD5 pentru cele 2 fisiere
    - apoi compara hash-urile obtinute si lereturneaza in forma de pointer array de tip int de dimensiunea 2 pe pozitia 0 fiind numarul de bytes egali din hash-ul SHA256 si pe pozitia 1 numarul de bytes egali pentru hash-ul MD5

1. Observatii
   1. In directorul tests se cum am mentionat si mai sus se afla o lista de teste

In main se itereaza prin acete teste si se ruleaza pe fiecare pereche de file1.in file2.in din ierarhiea de directoare si se apeleaza functia **hash\_and\_compare** care returneaza numarul de bytes care difera in hash-urile SHA256 respective MD5 peste acele fisiere raspunsul scriindu-se in fisierul test\_result.txt in urmatorul format:

test0

tests/0/file1.in vs tests/0/file2.in

SHA256 equal bytes : 0

MD5 equal bytes : 0

1. Modul de apelare: se apeleaza fisierul MakeFile care compileaza si executa main.cpp sau se ruleaza executabilul main din acelasi director