Criptare si decriptare

Proiectul are urmatoarea structura :

1. Zona bilbiotecilor, macrodefinitiilor si a definii de structuri

2. Functia int main()

3. Functia ScrieMatrice ()

4. Functia Citeste Matrice()

5. Functia CitesteHeader()

6. Functia Criptare()

7. Functia XORSHIFT32 ()

8. Functia Durstenfield ( )

9. Functia Decriptare()

10. Functia FrecventaCanalR()

11. Functia FrecventaCanalG()

12. Functia FrecventaCanalB()

13. Functia frecventa()

14. Functia FunctieCriptare()

15. Functia TestChi()

16. Functia FunctieDecriptare();

1. Zona bilbiotecilor, macrodefinitiilor si a definii de structuri

- definesc ( #define ) dimensiunea Headerului - 54

- folosesc pragma pack (1) pentru a elimina paddingul din structura ce va definita in continuare

- definesc o structura (Header) in care se pastreaza toate tipurile pentru campurile din header ,precizandu-se rolul lor ( b1 )

- definesc o structura (BMPMatrix) pentru a pastra pixelii din imaginile ce urmeaza a fi prelucrate

2. Functia int main()

- initializez si citesc de la tastatura numele fisierelor pe care urmeaza sa le folosesc in program

- daca nu sunt gasite opresc executia programului afisand un mesaj corespunzator

- declarare matrice de pixeli in forma liniarizata ( vector ) pentru imaginea care urmeaza sa fie prelucrata

- apelez functia CitesteHeader () (5)

- initializez dimensiunile matricei ( informatii care se gasesc in header )

- calculez nr de pixeli in dim si paddingul in padding

- apelez functia CitesteMatrice()(4) pentru I

- apelez TestChi ()(15) pentru I ( imaginea din sursa, necriptata)

- apelez functia FunctieCriptare()(14)

- apelez, din nou, TestChi()(15) pentru imaginea criptata

- apelez FunctieDecriptare()(16)

- eliberez toata memoria alocata dinamic

3. Functia ScrieMatrice ()

-salveaza in memoria externa, la o destinatie data, un pointer de tip BMPMatrix ( vector), precedat de o variabila de tip Header ( headerul imaginii si imaginea in pixeli )

- scrie imaginea invers

- tine cont de padding

4. Functia CitesteMatrice()

- parseaza headerul unei imagini cu vectorul buff[]

- citeste pixel cu pixeli matricea, luand in calcul paddingul

- o rastoarna si o incarca in memoria interna sub forma liniarizata ( de vector )

- returneaza un pointer simplu catre vectorul de pixeli citit

5. Functia CitesteHeader()

- citeste o variabila de tip header ( headerul unei imagini )

- returneaza variabila citita

6. Criptare()

-aloc memorie pt 2 matrici liniarizate : P si C, care au aceeasi dimensiune

- apelez XORSHIFT32()(7)

- apelez Durstenfiels()(8)

- aplic permutarea generata la pasul anterior

- xorez fiecare canal, al fiecarui pixel din vectorul de tip BMPMatrix, P, dupa formula si pastrez rezultatul in vectorul ( alocat la inceput) C

- eliberez memoria pentru tot ce a fost alocat dinamic in functie si retunez tabloul obtinut, C

7. XORSHIFT32 ()

- folosind generatorul de numere pseudo-aleatoare, xorshift32, returneaza un vector care pastreaza n-1 astfel de numere si , pe prima pozitie, seedul

8. Durstenfield ( )

- aloc memoria pt un tablou unidimensional cu n elemente

- il initializez astfel : v[i] = i, oricare ar fi i in intervalul [0, n)

- aplic algoritmul lui Dustenfield pe aceasta permutare, facand astfel un shuffle folosind ca numere aleatoare a doua jumatate a tabloului generat cu XORSHIFT32()(7),asupra carora aplic modul ( indicele pe care ma aflu + 1 )

9. Decriptare()

- aloc memorie pt 2 matrici liniarizate si un vector : P si C, care au aceeasi dimensiune si inv\_perm ( permutarea inversa)

- apelez XORSHIFT32()(7)

- apelez Durstenfiels()(8)

- creez permutarea inversa

- xorez fiecare canal, al fiecarui pixel din vectorul de tip BMPMatrix, I, dupa formula si pastrez rezultatul in vectorul ( alocat la inceput) C

- aplic permutarea inversa generata anterior obtinand tabloul P

- eliberez memoria pentru tot ce a fost alocat dinamic in functie si retunez tabloul obtinut, P

10.FrecventaCanalR()

-calculeaza , dupa formula data, frecventa pentru canalul rosu al pizelilor din matricea transmisa ca parametru si returneaza rezultatul

11.FrecventaCanalG()

-calculeaza , dupa formula data, frecventa pentru canalul rosu al pizelilor din matricea transmisa ca parametru si returneaza rezultatul

12.FrecventaCanalB()

-calculeaza , dupa formula data, frecventa pentru canalul rosu al pizelilor din matricea transmisa ca parametru si returneaza rezultatul

13.frecventa()

- are rolul de a afisa pe ecran valorile frecventei fiecarui canal de culor al vectorului de pixeli dat

- apelez FrecventaCanalR()(10) pentru matricea data

- apelez FrecventaCanalG()(11) pentru matricea data

- apelez FrecventaCanalB()(12) pentru matricea data

14.FunctieCriptare()

- declara o matrice de pixeli liniarizata - vector si un header

- apelez CitesteHeader ()(5)

- initializez dimensiunile imaginii si calculez paddingul, nr de pixeli

- apelez CitesteMatrice()(4)

-apelez Criptare()(6)

- apelez ScrieMatrice()(3) pentru I ( care este acum criptata )

-eliberez memoria lui I

15.TestChi()

- declara o matrice de pixeli liniarizata - vector si un header

- apelez CitesteHeader ()(5)

- initializez dimensiunile imaginii si calculez paddingul, nr de pixeli

- calculez frecventa estimata

- apelez frecventa() (13)

- eliberez memoria folosita pentru I

16.FunctieDecriptare()

- declara o matrice de pixeli liniarizata - vector si un header

- apelez CitesteHeader ()(5)

- initializez dimensiunile imaginii si calculez paddingul, nr de pixeli

- apelez CitesteMatrice()(4)

-apelez Criptare()(6)

- apelez Decriptare()(9)

- apelez ScrieMatrice()(3) pentru I ( care este acum decriptata )

-eliberez memoria lui I

Bibliografie :

(b1) https://engineering.purdue.edu/ece264/17au/hw/HW15

(b2) EnuntProiect.pdf