Criptografia si procesarea digitala a imaginilor

Criptografie:

1. Generatorul de numere XORSHIFT32
   - ANTET: unsigned int * XORSHIFT32(unsigned int seed, unsigned int n)
   - Definesc o structura numita pixel care retine cele trei culori
   - Primesc ca parametrii samanta si numarul de pixeli
   - Aloc memorie pentru un vector de numere random
   - Creez numerele si returnez un pointer catre vector
2. Functia de incarcare a unei imagini in memoria interna sub forma liniarizata
   - ANTET: pixel* incarca_imagine (char *nume_fisier_sursa)
   - Deschid fisierul si citesc marimile, calculez padding-ul
   - Aloc memorie pentru un vector de pixeli
   - Citesc in doua for-uri pixelii si ii pun in vector dupa formula
   - Indice_vector = indice_linie * numar_coloane + indice_coloana
3. Functia de salvare in memorie a imaginii
   - ANTET: void salvare_imagine(char *nume_fisier_sursa, char *nume_fisier_destinatie, pixel *C)
   - copiez header-ul, apoi pun elementele vectorului liniarizat
4. Functia de criptare
   - ANTET: void criptare(char *nume_fisier_sursa, char *nume_fisier_destinatie, char *cheie_secreta)
   - Citesc cheile, apelez XORSHIFT32, incarc imaginea
   - Fac permutarea conform algoritmului lui Durstenfeld

- Xor-ez pixelii si dupa salvez imaginea creata

5. Functia de decriptare
   - ANTET: decriptare(char *nume_fisier_sursa, char *nume_fisier_destinatie, char *cheie_secreta)
   - Citesc cheile, apelez XORSHIFT32, incarc imaginea
   - Xor-ez pixelii si fac permutarea, apoi salvez imaginea
6. Functia pentru calcularea valorilor chi patrat
   - void chi_patrat (char *nume_fisier)
   - calculez feceventele pixelilor de pe fiecare canal de culoare
   - calculez frecventa estimate teoretic
   - calculez valorile testului conform formulei si le afisez

Template maching

Functii si tipuri de date suplimentare:

- Folosesc functia gray_scale din documentatie
- Functia care calculeaza intensitatea unui sablon
- Functie care calculeaza intensitatea unei ferestre
- Functie care calculeaza deviatia unui sablon
- Functie care calculeaza deviatia unei ferestre
- Folosind formula si functia de mai sus, calzulez corelatia
- Functie care calculeaza daca doua ferestre se suprapun
- Definesc o structure numita fereastra cu doi indici
- Definesc o structura numita detective care retine corelatia, o culoare de tip pixel, o fereastra si existenta sa teoretica printr-o variabila cu valorile 0/1

- O functie comparator pentru cand vom sorta cu qsort
- Doua functii de incarcare si salvare care lucreaza cu vectori liniarizati de sus in jos

7. Functia de template maching
   - ANTET: void template_maching (char *nume_imagine, char *nume_sablon, double ps, pixel color, unsigned int *nr, detectie *D)
   - Creez un vector de detectii unde salvez ferestrele care au o corelatie mai mare decat pragul 0.5 cu sablonul trimis ca parametru, salvez culoarea ferestrei, corelatia si setez existenta pe 1
8. O functie care deseneaza o fereastra de o anumita culoare
   - ANTET: void contur (pixel *I, pixel C, fereastra f, unsigned int inaltime, unsigned int latime, unsigned int latimeI)
9. Sortez vectorul de detectii folosind qsort
10. Functia de eliminare a non-maximelor
    - ANTET: void eliminare_non_maxime (detectie *D, unsigned int numar_detectii, char *nume_sablon)
    - Iau doua for-uri si daca doua detectii exista si se suprapun setez existenta detectiei din dreapta la 0
11. Functia main
    - Citesc numele imaginilor
    - Apelez functile de criptare, decriptare, chi patrat si gray scale
    - Pentru fiecare culoare si sablon pe care le citesc, apelez functia template_maching
    - Elimin non maximele apeland functia
    - Pentru fiecare detectie existenta, fac un contur folosind fereastra detectiei, de culoarea gasita in detectie

Rezultatul final: