Functii utilizate

1. Funcția pentru generarea numerelor pseudo-aleatoare:

uint32_t xorshift32(uint32_t state)

Realizează generarea de numere pseudo-aleatoare pornind de la o valoare inițială, seed.

2. Funcții pentru încărcarea în memoria internă a unei imagini:

imagine citire(char* cale)

Funcțiile realizează citirea header-ului, respectiv a pixelilor imaginii.

3. Funcție pentru încărcarea în meoria externă a unei imagini:

void afisare(char* cale,imagine aux)

Se va copia imaginea altaturi de header pe un suport extern. Se utilizează padding-ul

4. Funcție pentru returnarea al i-lea byte al unui număr

unsigned char nth(uint32_t number,int n)

Tipul de date uint32_t este utilizat în lucrul cu biții, deoarece nu mai este necesar să detectăm lungimea unui long, int sau unsigned int

5. Funcție pentru criptarea imaginii

uint32_t* cript_a(imagine aux)

uint32_t *cript_b(uint32_t *a,imagine aux)

Pixel* cript_c(uint32_t *v,imagine aux)

Pixel* cript_d(Pixel* p2,uint32_t *v,imagine aux)

Se deschide fișierul cu imaginea ce dorim să o criptăm. Dacă fișierul nu poate fi găsit, se afișează un mesaj de eroare. Se alocă dimensiunea pentru înățime și lățime, se creează un vector pixels care conține cele 3 canale de culoare (roșu, verde, albastru), iar apoi folosind xorshift se generează numerele pseudo-aleatoare. Se construiesc permutărie pixelilor, iar apoi se substituie fiecare pixel cu valoarea criptată, urmând a se contrui imaginea intermediară.

6. Funcție pentru decriptarea imaginii

Pixel* decript_c(Pixel* c,uint32_t *v,imagine aux)

Pixel* decript_d(Pixel *p,uint32_t *v,imagine aux)

Se deschide fișierul cu imaginea ce dorim să o criptăm. Dacă fișierul nu poate fi găsit, se afișează un mesaj de eroare. Se alocă dimensiunea pentru înățime și lățime, se creează un vector pixels care conține cele 3 canale de culoare (roșu, verde, albastru), iar apoi folosind xorshift se generează numerele pseudo-aleatoare. Se construiesc permutărie pixelilor, iar apoi se construiește vectorul cu permutări inverse. Se permute pixelii înapoi, iar apoi se reconstruiește imaginea inițială.

7. Funcție pentru realizarea testului chi pătrat

void testchi(imagine aux){

Se deschide fișierul cu imaginea ce dorim să o criptăm. Dacă fișierul nu poate fi găsit, se afișează un mesaj de eroare. Se alocă dimensiunea pentru înățime și lățime, iar apoi inițializăm vectorii de frecvență cu valoarea nulă. Se calculează frecvențele, iar apoi calculăm testul chi pătrat.