

## **Functii folosite:**

- unsigned int\* XORSHIFT32(unsigned int seed, int n)
  - Genereaza numere intregi fara semn pe 32 de biti
  - Va incepe de la valoarea pe care o are seed si se va opri dupa generarea a (n-1) numere
- pixel\*\* matrice(int W, int H)
  - Alocare dinamica a unei matrice cu W coloane si H linii
- pixel\*\* citire(FILE \*f, int W, int H)
  - Salveaza in matricea de tip pixel fiecare canal de culoare din imaginea citita din fisier
- pixel\* liniarizare(int W, int H, pixel \*\*I)
  - Liniarizeaza matricea I in care este salvata imaginea
- int\* Durstenfeld(int W, int H, unsigned int \*R)
  - Genereaza o permutare aleatoare
  - R contine numerele aleatoare generate de functia XORSHIFT32
- pixel\* permutare(pixel \*L, int H, int W, unsigned int \*R, int \*p)
  - Permuta pixelii imaginii L conform permutarii aleatoare returnata de functia Durstenfeld
- pixel\* inversa(int W, int H, pixel \*C, int \*p)
  - Permuta pixelii imaginii criptate C conform inversei permutarii aleatoare salvata in p
- pixel\* criptare(pixel \*P, unsigned int \*R, unsigned int SV, int W, int H)
  - Se cripteaza imaginea specificata
  - Returneaza imaginea criptata prin vectorul C
  - Parametrii functiei sunt permutarea "P", numerele aleatoare "R", cheia secreta "SV", latimea imaginii "W" si inaltimea imaginii "H"
- pixel\* decriptare(pixel \*C, unsigned int \*R, unsigned int SV, int W, int H)
  - Se decripteaza imaginea
  - Returneaza imaginea decriptata prin vectorul D
  - Parametrii functiei sunt imaginea criptata liniarizata "C", numerele aleatoare "R", cheia secreta "SV", latimea imaginii "W" si inaltimea imaginii "H"

- void creare(FILE \*fin, FILE \*fout, pixel \*C, int H, int W)
  - Salveaza imaginea din C in memoria externa
- pixel\* frecventa(pixel \*L, int W, int H)
  - Calculeza frecventa fiecarei culori din imagine
- void test\_chi(int m, int n, pixel \*I)
  - Afiseaza numarul pixelilor din cele 3 canale de culoare dintr-o imagine liniarizata "I"
  - m reprezinta numarul de linii al matricei si n numarul de coloane
- void grayscale\_image(char\* nume\_fisier\_sursa, char\* nume\_fisier\_destinatie)
  - Transforma imaginea color in imagine alb-negru.
- double media\_intensitatilor(int i, int j, int n, int m, pixel \*\*I)
  - Furnizeaza media valorilor intensitatilor alb-negru a pixelilor din imaginea salvata in matricea I
  - i si j reprezinta coordonatele punctului de start, iar n este numarul de linii si m numarul de coloane
- double deviatia\_standard(int i, int j, int n, int m, pixel \*\*I)
  - Furnizeaza deviatia standard a intensitatilor alb-negru a pixelilor din imaginea I
- pixel\*\* colorare(int i, int j, pixel \*\*C, int c)
  - Schimba culoarea pixelilor in functie de valoarea cifrei din c si o salveaza in matricea C
- detectii\* glisare(pixel \*\*I, int H, int W, int h, int w, pixel \*\*S, int c, detectii \*D, int \*k, double ps)
  - Gliseaza sablonul pe imaginea "decriptare2.bmp" si calculeaza corelatia dintre sablonul curent si continutul corespunzator al imaginii, dat de dimensiunea sablonului.
  - h si w sunt inaltimea si latimea sablonului S
  - in vectorul D salveaza detectiile care au corelatia mai mare decat pragul ps
- int cmp(const void \*a, const void \*b)
  - Se foloseste pentru sortarea descrescatoare a detectiilor in functie de corelatie.

➤ void suprapunere(detectii \*\*D, int \*k)

- Calculeaza suprapunerea spatiala dintre doua detectii D[i] si D[j]
- Daca suprapunerea spatial este mai mare decat 0.2 atunci se sterge detectia D[j]
- Utilizeaza functia colorare asupra detectiilor ramase