

## PROGRAMARE PROCEDURALA

### DOCUMENTATIE

**Introducere :** Numele meu este *Tcaciuc Iulian-Marius* si in acest fisier Word este prezentata documentatia proiectului de anul acesta. Sunt prezentate 2 parti : CDT (criptare, decriptare, test) si Template.

#### Partea I - > CDT

Pentru partea I am ales o implementare prin structuri. Programul prezinta 2 structuri ( pixel si bmp). Structura pixel este reprezentata de cele 3 culori (r,g,b) iar structura bmp memoreaza caracteristicile unei imagini ( lungime, latime, pixeli, header si padding). Programul prezinta 8 functii: xorshift32 , incarcareBMP , salvareBMP , xoor1, xoor2 , criptare, decriptare, test.

**Functia xorshift32 :** Genereaza numere pseudo-aleatoare pe 32 de biti -> <https://en.wikipedia.org/wiki/Xorshift>

**Functia incarcareBMP :** In aceasta functie realizam citirea caracteristicilor unei imagini pe care le memoram in variabila v precum si liniarizarea imaginii transmisa ca parametru "numeBmp". Aceasta incarca in memoria interna imaginea "numeBmp". Functia returneaza "v" in care avem liniarizarea si caracteristicile imaginii "numeBmp".

**Functia salvareBMP :** In aceasta functie realizam salvarea in memoria externa a imaginii numeBmp scrisa in forma liniarizata in "v.p". Imaginea "numeBmp" se salveaza in imaginea "numeDestinatie" trimisa ca parametru. Functia returneaza variabila v.

**Functia xoor1 :** In aceasta functie realizam "xor" intre 2 pixeli . Returneaza o variabila "c" de tip "struct pixel" dupa xorarea dintre pixelii y si z transmisi ca parametrii.

**Functia xoor2 :** In aceasta functie realizam "xor" intre 1 pixel si un numar intreg. Returneaza o variabila "c" de tip "struct pixel" dupa xorarea dintre intregul x si pixelul y.

**Functia criptare :** In aceasta functie realizam criptarea imaginii "numeBmp". Imaginea criptata va fi stocata in vectorul criptat, care este de tip "struct pixel" si salvata in "bmpCriptat". Functia returneaza vector liniarizat in care avem imaginea criptata. Functia are 3 parametrii : imaginea initiala ( numeBmp) , imaginea criptata (bmpCriptat) si fisierul unde avem cheia secreta (secretKey)

**Functia decriptare:** In aceasta functie realizam decriptarea imaginii "bmpCriptat" . Imaginea decriptata va fi stocata in vectorul decriptat, care este de tip "struct pixel" si salvata in "bmpDecriptat". Functia are 4 parametrii : numeBmp,bmpCriptat,secretKey si bmpDecriptat in care avem decriptata imaginea "bmpCriptat".

**Functia test:** In aceasta functie realizam rezolvarea cerintei 6 si anume aflarea valorilor testului  $x^2$  pentru imaginea criptata "bmpCriptat" pe fiecare canal de culoare (r,g,b). Prezinta 3 parametrii : numeBmp , bmpCriptat si secretKey explicate mai sus.

## Partea II - > Template

Pentru partea II am ales o implementare prin transmitere de parametrii deoarece aveam deja 2 structuri. Programul prezinta 2 structuri ( pixel si detectie). Structura pixel este reprezentata de cele 3 culori (r,g,b) iar structura detectie contine 3 campuri : cor( corelatie) , x si y (coordonatele) si cul ( tip struct pixel ). Programul prezinta 12 functii : incarcareBMP, salvareBMP, grayscaleBMP, colorareBMP, corelatie, templatematching, cmp, supraScriere, minim, maxim, arie, eliminareNonmaxime.

**Functia incarcareBMP** : In aceasta functie realizam citirea caracteristicilor imaginii "numeBmp" pe care o memoram in vectorul liniarizat "v" de tip "struct pixel". Functia transmite ca parametrii caracteristicile imaginii "numeBmp" (lungime, latime , header ) si vectorul cu liniarizarea de tip pixel "v".

**Functia salvareBMP**: In aceasta functie realizam salvarea in memoria externa a imaginii "numeBmp" in "numeDestinatie" ambele trimise ca parametrii. Functia mai transmite ca parametrii si caracteristicile imaginii (lungime, latime, header) si vectorul in care avem liniarizarea.

**Functia grayscaleBMP**: In aceasta functie realizam transformarea imaginii pe care o avem in primul parametru in pixeli gri. Functia are ca parametrii vectorul cu liniarizarea si dimensiunea (lungime\*latime) stocata in "dimBMP".

**Functia colorareBMP**: In aceasta functie realizam colorarea imaginii pe care o avem stocata in primul parametru. Functia mai are ca parametrii si dimensiunile imaginii, culoarea necesara pentru colorare si coordonatele de unde incepem sa coloram chenarul.

**Functia corelatie**: In aceasta functie calculam corelatia dupa formula data in enuntul proiectului. Aceasta functie returneaza corelatia prin "sumaCorelatie". Functia are ca parametrii vectorul "vectorBMP" in care avem imaginea mare "numeBmp" si dimensiunile respective ( lungime, latime), vectorul "vectorSablon" in care avem imaginile cu cifre (11 \* 15 ) , medieSGrayscale si deviatieSablon ale sablonului mic din enuntul proiectului si coordonatele x,y.

**Functia templatematching**: In aceasta functie realizam crearea vectorului de detectii "v" transmis ca parametru si colorarea vectorului initializat in program "detect". Functia mai prezinta ca parametrii imaginea "numeBmp", imaginea care va fi colorata "numeDetectie", sablonul "numeSablon" cu cifrele care se vor apela in main pe rand, dimensiunile imaginii(lungime,latime), dimensiunea vectorului de detectie (dimBMP), culoarea care se va apela pe rand in main si pragul de corelatie (pragS).

**Functia cmp** : Functie de comparare necesara pentru qsort.

**Functia supraScriere** : Verifica daca doua imagini se suprapun folosind o formula gasita pe [geeksforgeeks](https://www.geeksforgeeks.org/find-two-rectangles-overlap/)  
<https://www.geeksforgeeks.org/find-two-rectangles-overlap/>

**Functia maxim** : Returneaza maximul dintre 2 numere transmise ca parametrii.

**Functia minim**; Returneaza minimul dintre 2 numere transmise ca parametrii.

**Functia arie** : Calculeaza suprafata care se intersecteaza conform formulei din enunt.

**Functia eliminareNonMaxime** : In aceasta functie se elimina nonmaximele. Prezinta ca parametrii vectorul de detectii v, si dimensiunea acestuia.

*In main apelam templatematching pentru fiecare cifra si culoare, apelam functia qsort pentru sortarea descrescatoare , eliminam nonmaximele si salvam imaginea "bmpFinal" in fisierul binar.*