

# Documentatie . Programare procedurala - Proiect.

## Badea Adrian- Catalin

Ianuarie 2019

---

### Continutul arhivei :

Proiectul contine : Documentatia (*Documentatie.pdf*)

Fisierul sursa, (*main.c*), echivalent task-ului 11

Diferitele header-uri, (*bmp.h, vector.h, encryption.h, templatematching.h*)  
programelor :

Pattern-urile (*cifra0.bmp, cifra1.bmp, ..., cifra9.bmp*);

Imaginele in Bitmap (*peppers.bmp, test.bmp*)

Fisierul ce contine cheile secrete (*secret\_key.txt*)

---

### 1. Structuri :

- *header* : Stocheaza datele despre un header, un sir de 54 de bytes, echivalent primilor bytes dintr-o image de tip Bitmap.
- *bmpimage* : contine datele necesare procesarii unei imagini : o structura de tip header, un vector alocat dinamic pentru fiecare canal de culoare si trei variabile de tip int: inaltimea, latimea si size-ul imaginii
- *pix* : stocheaza date despre fiecare canal de culoare (blue, green, red)
- *corr\_elem* : contine coordonatele pe Ox si pe Oy ale unei ferestre, corelatia dintre aceasta si un sablon dat si cifra sablonului.
- *vector* : stocheaza datele necesare pentru stocarea-corelatiilor. Astfel, contine : un vector alocat dinamic, cu element de tip *corr\_elem*. De asemenea, contine dimensiunea si capacitatea acestuia, pentru a facilita realocarea dinamica).

### 2. Informatii despre functiile implementate :

- *Encryption.h*, contine *xorshift32*, *createsequence*, *createpermutation*, *xoratepixelint*, *xoratepixels*, *encrypt\_image*, *decrypt\_image*, *chi\_function*, pe care le vom explica in continuare:

- *xorshift32* : Primeste ca parametru un numar generat random, sau valoarea de start si returneaza urmatorul numar din secventa, folosind principiile xorshift32.
- *createsequence* : Primeste ca parametru cheia secreta din fisierul *secret\_key.txt*, si lungimea secventei de numere si creeaza aceasta secventa, care va fi folosita ulterior la crearea permutarii. Returneaza un pointer catre secventa alocata dinamic pe memorie.
- *createpermutation* : Creeaza o permutare folosind algoritmul lui Durstenfeld, generata random pe baza primii jumatați din secventa randomizata. Aceasta are ca parametru secventa random si dimensiunea permutarii.
- *encrypt\_image* : Crijteaza o imagine BMP folosind algoritmul de criptare prezentat. Primeste ca parametru denumirile fisierlor in care apare imaginea initiala, unde va scrie imaginea dupa criptare si unde se afla cheia secreta.
- *decrypt\_image* : Decrijteaza o imagine BMP crijata folosind algoritmul de criptare prezentat, prin functia *encrypt\_image* . De asemenea, primeste ca parametru denumirile fisierlor in care apare imaginea initiala, unde va scrie imaginea dupa criptare si unde se afla cheia secreta.

January 4, 2019

- *bmp.h* : Header-ul contine structura unei imagini bitmap, si functiile aferente structurii, de scriere si de citire, pe care le vom explica dupa cum urmeaza:
- *createbmp* : Contine crearea imaginii de tip bitmap, dupa ce sunt extrase inaltimea si latimea din header-ul acesteia. In continuare se aplica linearizarea matricei pentru citire din fisier.
- *matrix\_linearisation* : Realizarea citirii dintr-un fisier dat. Citeste din fisierul de tip bmp header-ul, inaltimea, latimea
- *save\_linearisation* : Primeste ca parametru o imagine bmp si un fisier in memoria externa si salveaza in fisierul dat imaginea. Copiaza header-ul structurii si apoi pixelii in ordinea in care se afla natural in memorie
  - *templatematching.h* : Header-ul contine structura unei imagini bitmap, si functiile aferente structurii, de scriere si de citire, pe care le vom explica dupa cum urmeaza:
- *grayscaleimage* : Primeste ca parametru o imagine de tip bitmap si ii transforma intensitatile pixelilor asa cum este prezentat in descrierea proiectului, pentru a crea o imagine scalata in nuante de gri.
- *template\_matching* : Primeste ca parametru imaginea in care cautam sabloanele si un sablon si returneaza o structura de tip vector, in care se gasesc corelatiile mai mari ca un anumit prag, dat constant 0.5
- *contouring* : Primeste ca parametru un pointer catre o imagine, si datele unei corelari, si creeaza, in functie de aceste date, un dreptunghi care contine cifra unde s-a facut corelarea in imagine.
- *compare* : Functia comparator compatibila cu qsort-ul, care ordoneaza detectiile in functie de corelatia lor.
- *sortingthematches* : Functia sorteaza un tablou de detectii in ordinea descrescatoare a corelatiilor detectiilor. Primeste ca parametru un pointer catre un vector de detectii
- *nonmax\_elimination* : implementeaza algoritmul de eliminare a non-maximelor, asa cum este prezentat in descrierea proiectului.
- *overlapped\_enough* : Implementeaza criteriul de suprapunere descris in proiect, avand un prag minim 0.2
  - *vector.h* : Header-ul contine structura vectorului care se realoca dinamic, binar, in functie de numarul de elemente de care are nevoie.
  - *Constante* :
- *PIX\_NUMBER* = 165
- *PRAG* = 0.5
- *OP\_PRAG* = 0.2