

Proiect Programare Procedurala

Pentru rezolvarea proiectului au fost folosite urmatoarele structuri:

```
typedef struct ImgData
{
    unsigned char header[54];
    unsigned int W;
    unsigned int H;
    unsigned int size;
    unsigned char *data;
} ImgData;

typedef struct dreptunghi
{
    int stanga;
    int dreapta;
    int sus;
    int jos;
} dreptunghi;

typedef struct matchData
{
    int cifra;
    int x;
    int y;
    double corr;
} matchData;
```

Structura “ImgData” memoreaza urmatoarele datele specifice unei imagini de tip “BMP”:

- Header-ul imaginii alocat dinamic – ‘header’
- Latimea – ‘W’
- Intaltimea – ‘H’
- Dimensiunea in pixeli a imaginii – ‘Size’
- Informatii stocate dinamic (culorile de pe canalele R,G,B) specifice fiecarui pixel din imagine – ‘data’

Structura “dreptungi” este folosita in calcularea ariei rezultata din suprapunerea a doua dreptunghiuri, memorand indicii varfurilor “sus-stanga” si “jos-dreapta” ale unui dreptunghi.

Structura “matchData” memoreaza informatii despre o detectie generata de algoritmul “Template matching”:

- Indicii detectiei raportat la imaginea scanata – ‘X’, ‘Y’
- Cifra al carei sablon a generat aceasta detectie
- Corelatia dintre sablon si imaginea scanata – ‘corr’

Funcțiile folosite in rezolvarea modulului de criptare/decriptare:

```
void BMPtoImgData(char const filename[], ImgData *image)
```

- Primește ca parametrii: calea unei imagini, pointer catre o variabila de tip “ImgData”.
- Deschide o imagine de tip “BMP” din memoria externa si transmite informatiile (pixeli, marime, header) prin al doilea parametru. Pixelii vor fi memorati dinamic in mod liniarizat.
- Nu returneaza ceva.

```
void ImgDatatoBMP(char const filename[], ImgData image)
```

- Primește ca parametrii: calea unei imagini, o variabila de tip “ImgData”.
- Converteste informatiile aferente unei imagini de tip “BMP” din memoria interna in memoria externa (o imagine cu formatul “.bmp” specificata in primul parametru).
- Nu returneaza ceva.

```
void XORSHIFT32(unsigned int **v, unsigned int size, unsigned int seed)
```

- Primește ca parametrii: un pointer catre un array unidimensional de numere intregi fara semn, marimea array-ului (numarul de elemente), un numar intreg fara semn.
- Genereaza un array unidimensional de numere intregi fara semn aleatoare, raportate la samanta primita prin ultimul parametru. Array-ul se transmite prin primul parametru.
- Nu returneaza ceva.

unsigned int * Genereaza_Permutare(**unsigned int ***rand, **unsigned int** size)

- Primește ca parametrii: un array unidimensional de numere întregi fara semn, marimea array-ului (numarul de elemente).
- Genereaza un array unidimensional alocat dinamic de permutari, generat din numerele aleatoare transmise prin parametru.
- Returneaza un pointer catre array-ul de permutari generat.

void Inverseaza_Permutare(**unsigned int **v**, **unsigned int** size)

- Primește ca parametrii: un pointer catre un array unidimensional de numere întregi fara semn, marimea array-ului (numarul de elemente).
- Transforma array-ului de permutari transmis prin parametru intr-un array de permutari inverse.
- Nu returneaza ceva.

void GetBytes(**unsigned int** x, **unsigned char **byte**)

- Primește ca parametrii: un numar întreg fara semn, un pointer catre un array de “char-uri” fara semn.
- Memoreaza in array-ul transmis prin ultimul parametru, descompunea numarului întreg fara semn transmis prin primul parametru.
- Nu returneaza ceva.

void PixelXOR(**unsigned char **v**, **unsigned char *a**, **unsigned char *b**, **unsigned char *c**)

- Primește ca parametrii: un pointer catre un array de “char-uri” fara semn, alte 3 array-uri de “char-uri” fara semn.
- XOR-eaza cele 3 array-uri (element cu element) si storeaza rezultatele in primul array.
- Nu returneaza ceva.

void Cripteaza(**char const** in_filename[], **char const** out_filename[], **char const** secretCode_filename[])

- Primește ca parametrii: calea unui fisier de intrare (image BMP), calea unui fisier de iesire (image BMP), calea unei fisier care contine codurile secrete necesare criptarii.

- Cripteaza imaginea “de intrare” folosindu-se de codurile secrete din fisierul specificat in ultimul parametru si o exporteaza ca imaginea “de iesire” specificata in al doilea parametru.
- Nu returneaza ceva.

```
void Decripteaza(char const in_filename[], char const out_filename[], char const secretCode_filename[])
```

- Primeste ca parametrii: calea unui fisier de intrare (imagine BMP), calea unui fisier de iesire (imagine BMP), calea unei fisier care contine codurile secrete necesare decriptarii.
- Decripteaza imaginea “de intrare” folosindu-se de codurile secrete din fisierul specificat in ultimul parametru si o exporteaza ca imaginea “de iesire” specificata in al doilea parametru.
- Nu returneaza ceva.

```
void CalculeazaHistograma(char const in_filename[])
```

- Primeste ca parametrii: calea unui fisier (imagine BMP).
- Calculeaza valorile testului χ^2 pentru imaginea BMP, transmisa prin parametru, pe fiecare canal de culoare, dupa care le afiseaza pe ecran.
- Nu returneaza ceva.

Functii folosite in rezolvarea modului de recunoastere de pattern-uri intr-o imagine:

```
void GrayScale(ImgData *img)
```

- Primeste ca parametrii: un pointer catre o variabila de tip “ImgData”.
- Converteste “imaginea” stocata in variabila din parametru intr-una grayscale (tonuri de gri).
- Nu returneaza ceva.

```
double MedieIntensitatiSablon(ImgData img)
```

- Primeste ca parametrii: o variabila de tip “ImgData” (sablonul folosit pentru detectii).

- Calculeaza media valorilor intensităților grayscale a pixelilor “imaginii” din parametru.
- Returneaza rezultatul ca un numar real cu precizie dubla.

double DeviatieIntensitatiSablon(ImgData img)

- Primeste ca parametrii: o variabila de tip “ImgData” (sablonul folosit pentru detectii).
- Calculeaza deviația standard a valorilor intensităților grayscale a pixelilor “imaginii” din parametru.
- Returneaza rezultatul ca un numar real cu precizie dubla.

double MedieIntensitatiImagine(ImgData image, ImgData tmp, **int** n, **int** m)

- Primeste ca parametrii: o variabila de tip “ImgData” (imaginea pe care se gliseaza sablonul), o alta variabila de tip “ImgData” (sablonul care se gliseaza pe imagine), doua numere intregi reprezentand indicii/pozitia in functie de care se gliseaza sablonul peste imagine.
- Calculeaza media valorilor intensităților grayscale a pixelilor “imaginii” din parametru, raportata la marimea “sablonului” si la indicii de unde se gliseaza “sablonul”.
- Returneaza rezultatul ca un numar real cu precizie dubla.

double DeviatieIntensitatiImagine(ImgData image, ImgData tmp, **int** n, **int** m)

- Primeste ca parametrii: o variabila de tip “ImgData” (imaginea pe care se gliseaza sablonul), o alta variabila de tip “ImgData” (sablonul care se gliseaza pe imagine), doua numere intregi reprezentand indicii/pozitia in functie de care se gliseaza sablonul peste imagine.
- Calculeaza deviația standard a valorilor intensităților grayscale a pixelilor “imaginii” din parametru, raportata la marimea “sablonului” si la indicii de unde se gliseaza “sablonul”.
- Returneaza rezultatul ca un numar real cu precizie dubla

void draw(ImgData *image, ImgData tmp, **int** x, **int** y, **char** R, **char** G, **char** B)

- Primeste ca parametrii: o variabila de tip “ImgData” (imaginea pe care se gliseaza sablonul), o alta variabila de tip “ImgData” (sablonul care se gliseaza pe imagine), doua numere intregi reprezentand indicii/pozitia in functie de care se gliseaza sablonul peste

image, alte 3 variabile de tip “char” reprezentand valorile unei culori pe canalele R, G, B.

- Deseneaza in “imagea” specificata in primul parametru, un dreptunghi de dimensiunea “sablonului”, de culoarea specificata in ultimii trei parametrii, incepand cu cei doi indici ‘x’, ‘y’.
- Nu returneaza ceva.

```
void TemplateMatch(matchData **matches, int *matchesCount, int cifra, char const image_filename[], char const tmp_filename[], double prag)
```

- Primeste ca parametrii: un pointer catre un array care va retine detectiile generate de functie, un numar intreg reprezentand numarul detectiilor, un numar intreg reprezentand cifra specifica sablonului cu care se genereaza detectiile, calea imaginii care va fi scanata, calea sablonului cu ajutorul caruia se vor genera detectiile, un numar real cu precizie dubla reprezentand acuratetea minima a unei detectii.
- Genereaza un array de detectii determinat de glisarea repetata a sablonului in toate pozitiile imaginii. Detectiile se considera valide daca acuratetea lor este peste pragul minim specificat in ultimul parametru. Array-ul, precum si numarul de detectii vor fi transmise prin parametrii aferenti.
- Nu returneaza ceva.

```
int cmp(void const *A, void const *B)
```

- Primeste ca parametrii: doi pointer catre doua variabile de format necunoscut (In proiect se vor transmite doua variabile de tip “matchData”).
- Compara cele 2 variabile in functie de corelatie.
- Returneaza:
 - 1, daca prima variabila are o corelatie mai mica decat a doua variabila.
 - -1, daca prima variabila are o corelatie mai mare decat a doua variabila.
 - 0, altfel

```
void SortDesc(matchData **matches, int count)
```

- Primeste ca parametrii: un pointer catre un array de elemente de tip “matchData” (detectii), un numar intreg reprezentand numarul de elemente.
- Sorteaza descrescator elemente array-ului in functie de coeficientul de corelatie. Array-ul modificat se transmite prin primul parametru.
- Nu returneaza ceva.

```
int min(int a, int b)
```

- Primește ca parametrii: doua numere intregi.
- Calculeaza minimul dintre cele doua numere.
- Returneaza rezultatul.

```
int max(int a, int b)
```

- Primește ca parametrii: doua numere intregi.
- Calculeaza maximul dintre cele doua numere.
- Returneaza rezultatul.

```
int intersectie(matchData m, matchData n, ImgData tmp)
```

- Primește ca parametrii: doua variabile de tip “matchData” (detectii), o variabila de tip “ImgData” (sablon).
- Calculeaza aria intersectiei dintre cele doua detectii.
- Returneaza rezultatul ca un numar intreg.

```
double suprapunere(matchData m1, matchData m2, ImgData tmp)
```

- Primește ca parametrii: doua variabile de tip “matchData” (detectii), o variabila de tip “ImgData” (sablon).
- Calculeaza coeficientul de suprapunere al celor doua detectii.
- Returneaza rezultatul ca un numar real cu precizie dubla cuprins intre 0.0 si 1.0 .

```
void RemoveNonMax(matchData **matches, int count, ImgData tmp, double prag)
```

- Primește ca parametrii: un pointer catre un array de elemente de tip “matchData” (detectii), un numar intreg reprezentand numarul de elemente, o variabila de tip “ImgData” (sablon), un numar real cu precizie dubla reprezentand un prag de selectie.
- Marcheaza detectiile din array care au un coeficient de suprapunere mai mare decat pragul primit ca parametru. Detectiilor corespunzatoare criteriului de selectie li se vor seta unul din idicii de pozitie (‘X’) la valoarea -1.
- Nu returneaza ceva.