

Proiect PP – Dec 2018

Simionescu Codrut Andrei – gr 133

Explicarea functiilor

(in ordinea aparitiei)

```
+void menu_C11() { ... }
```

Meniul programului: realizeaza interfata cu utilizatorul acestuia si alegerea functionalitaii dorite (criptare, decriptare, recunoastere).

```
+void encryptImage_C4(char *nume_fisier_sursa, char *nume_fisier_destinatie, char *secret_key) { ... }
```

Primește ca parametru numele unei imagini sursa și o criptează, salvând-o cu al doilea nume primit ca parametru, folosind cheia secretă din fișierul al cărui nume este transmis al treilea.

```
int getSecretKeys(char *nume_fisier_secret_key, unsigned int *R0, unsigned int *SV) { ... }
```

Deschide fisierul ce contine cheile secrete, le citeste si le transmite prin parametrii. Returneaza 1 daca fisierul a fost gasit, 0 in caz contrar.

```
unsigned int loadImageLin_C2(char *nume_fisier_sursa, unsigned char **header, struct pixel **L, unsigned int *latime_img, unsigned int *inaltime_img, unsigned int *padding, int okType) { ... }
```

Deschide imaginea cu numele transmis ca prim parametru si o citeste in memoria interna in forma liniarizata in vectorul de structuri L, alaturi de header ul, inaltimea si latimea acesteia, calculand si padding ul, daca este necesar. Returneaza dimensiunea acesteia, sau 0, in cazul in care imaginea sursa nu a fost gasita.

```
void xorshift32_C1(unsigned int **R, unsigned int n, unsigned int seed) { ... }
```

Genereaza o secventa aleatoare de n numere, folosind seed ul transmis, si o salveaza in vectorul R.

```
void pDurstenfeld(unsigned int **P, unsigned int n, unsigned int *R) { ... }
```

Genereaza permutarea Durstenfeld, folosind n numere din R, si o salveaza in P.

```
+void saveImageLin_C3(char *nume_fisier_destinatie, unsigned char *header, struct pixel *L, unsigned int *L, unsigned int latime_img, unsigned int inaltime_img, unsigned int padding){ ... }
```

Salveaza o imagine sub numele transmis prin parametru din memoria interna (in care se afla sub forma liniarizata in vectorul L) in memoria externa.

```
+void chiSquaredTest_C6(char *nume_fisier_testat){ ... }
```

Aplica testul chi squared pe o imagine al carui nume e transmis prin parametru si afiseaza valorile obtinute, atat pe ecran, cat si intr-un fisier text.

```
+void decryptImage_C5(char *nume_fisier_decriptat, char *nume_fisier_criptat, char *secret_key){ ... }
```

Primește ca al doilea parametru numele unei imagini sursa criptate și o decriptează, salvând-o cu primul nume primit ca parametru, folosind cheia secretă din fisierul al cărui nume este transmis al treilea.

```
+void inversaPerm(unsigned int **P, unsigned int n){ ... }
```

Primește și inversează permutarea P, de n elemente, primită ca parametru.

```
void patternMatching(char *nume_fisier_sursa, char *nume_fisier_rec, char *nume_fisier_sabloane, float ps) { ... }
```

Aplica tehnica de pattern matching pe imaginea al carui nume este transmis prin parametru, folosind pragul ps, si o salveaza cu cel de-al doilea nume transmis prin parametru.

```
unsigned int grayscaleImage(char* nume_fisier_sursa, char* nume_fisier_destinatie, int okType) { ... }
```

Primește numele unei imagini sursa si o salveaza in forma grayscale, sub al doilea nume primit ca parametru. Returneaza dimensiunea acesteia, sau 0, in cazul in care imaginea sursa nu a fost gasita.

```
struct detectie *templateMatching_C7(char *nume_fisier_sursa, char *nume_sablon, float prag, unsigned int *cnt, int nrSab) { ... }
```

Aplica operatia de template matching dintre o imagine si un sablon, folosind pragul dat, si returneaza vectorul detectiilor obtinute in urma verificarii, si nr acestora.

```
struct detectie windowMatch(struct pixel **MatI, struct pixel **S, int x, int y, unsigned int inaltime, unsigned int latime, int nrSab) { ... }
```

Verifica corelatia unei ferestre a imaginii cu un sablon, ambele transmise sub forma de matrice, si returneaza detectia obtinuta.

```
+void sortDetect_C9(struct detectie *arr, unsigned int n) { ... }
```

Sorteaza vectorul de detectii de n elemente descrescator, folosind qsort.

```
+int cmp_C9(const void *a, const void *b) { ... }
```

Functia de comparare transmisa qsort ului.

```
+void delete_non_max_C10(struct detectie *arr, unsigned int *n) { ... }
```

Primește vectorul de detectii de n elemente sortat descrescator si elimina detectiile care se suprapun cu cele de corelatie maxima.

```
+struct pixel **loadMatI(char *nume_fisier_sursa, unsigned char **header) { ... }
```

Returneaza matricea imaginii al carui nume e transmis prin parametru.

```
+void setColors(struct pixel **C) { ... }
```

Seteaza un vector de culori, cu care urmeaza sa fie desenate bordurile detectiilor.

```
+void drawBorder_C8(struct pixel **MatI, struct detectie fI, struct pixel C) { ... }
```

Deseneaza in imaginea transmisa ca matrice, bordura detectiei fI, cu culoarea C.

```
+void saveMatI(char *nume_fisier_dest, struct pixel **MatI, unsigned char *header) { ... }
```

Salveaza in memoria externa o imagine cu numele primit, transmisa sub forma de matrice.