1. Functia xorshift are ca parametrii tabloul unidimensional alocat dinamic R care reprezinta numerele pseudoaleatoare generate cu algoritmul xorshift, latimea imaginii initiale (latime_img), inaltimea imaginii initiale(inaltime_img), si valoarea initiala cu care este initializat primul numar(seed). Dupa initializare, pentru indicii de la 1 la latime_img * inaltime_img - 1, la fiecare iteratie aplicam xorshift pentru numarul anterior si stocam noul numar obtinut in tablou.

2. Functia incarcare_BMP are ca parametrii calea imaginii pe care vrem sa o stocam in forma liniara in memoria interna (nume_img), tabloul unidimensional alocat dinamic P in care va fi salvata, latimea imaginii(latime_img) si inaltimea imaginii(inaltime_img). Salvam latimea si inaltimea imaginii care se gasesc incepand cu octetul 18, respectiv 22 in header-ul imaginii, dupa care calculam padding-ul, alocam memoria pentru P si stocam "liniile" imaginii in ordine inversa.

3. Functia salvare_BMP are ca parametrii calea imaginii in care se va salva tabloul ce contine pixelii(nume_img), tabloul unidimensional P care contine pixelii, latimea imaginii(latime_img), inaltimea imaginii(inaltime_img) si header-ul imaginii. Calculam padding-ul si stocam "liniile" corespunzatoare tabloului unidimensional daca ar fi stocat in forma bidimensionala in ordine inversa(formatul BMP are liniile de pixeli "rasturnate") si adaugam octetii de padding corespunzatori.

4. Functia criptare_BMP are ca parametrii calea imaginii initiale(nume_img_initiala), calea imaginii criptate(nume_img_criptata), calea cheii secrete(nume_cheie_secreta), tabloul unidimensional P care contine pixelii imaginii initiale, latimea imaginii initiale(latime_img) si inaltimea imaginii initiale(inaltime_img). Stocam header-u imaginii initiale, dupa care generam numerele pseudoaleatoare cu functia xorshift, dupa care generam permutarea aleatoare cu algoritmul lui Durstenfeld, folosind numerele pseudoaleatoare din tabloul R modulo (i + 1) pentru a ramane in intervalul 0, i. Dupa aceea, folosindu-ne de un tablou auxiliar alocat dinamic (auxP), permutam pixelii imaginii dupa permutarea obtinuta si stocata in perm. Realizam apoi relatia de substitutie a pixelilor pentru fiecare culoare (rosu, verde, albastru), iar pixelul criptat va fi obtinut prin xorshiftarea pixelului criptat precedent cu pixelul initial curent si cu octetul corespunzator din numarul pseudoaleator curent. Salvam in imaginea criptata tabloul P modificat ce contine pixelii imaginii initiale criptati.

5. Functia decriptare_BMP are ca parametrii caile imaginii criptate, imaginii decriptate, si cheii secrete, tabloul de pixeli P, latimea si inaltimea imaginii. Se aplica acelasi rationament ca si la functia de criptare, numai ca acum vom calcula permutarea inversa a permutarii obtinute prin aplicarea algoritmului Durstenfeld, iar la relatia de substitutie pixelul decriptat va fi obtinut prin xorshiftarea pixelului criptat precedent cu pixelul criptat curent si octetul corespunzator din numarul pseudoaleator curent.

6. Functia valori_chi_square are ca parametrii calea imaginii pe care vrem sa facem testul, latimea si inaltimea ei. Calculam in cate un tablou frecventele pe fiecare canal de culoare a valorilor pixelilor, dupa care calculam frecventa estimata si aplicam formula pentru a obtine valoare chi square pentru fiecare canal de culoare.

7. Functia grayscale_image are ca parametrii tabloul de pixeli P, latiema si inaltimea imaginii a caror pixeli vrem sa ii transformam in gri, iar valoarea pentru fiecare canal de culoare a unui pixel va fi obtinuta insumand produsul dintre 0.299 si valoarea pe canalul rosu cu produsul dintre 0.587 si valoarea

pe canalul verde si cu produsul dintre 0.114 si valoarea pe canalul albastru.

8.Functia template_matching are ca parametrii tabloul P de pixeli al imaginii, latimea si inaltimea imaginii (latime_img si inaltime_img) pe care se face template matching, tabloul S de pixeli al sablonului, latimea si inaltimea sablonului (latime_sablon si inaltime_sablon), pragul ps, tabloul unidimensional "fereastra" in care vom stoca indicele liniei si coloanei elementului din stanga sus a fiecarei detectii, numarul de ferestre(detectii) gasite pentru sablon in imaginea stocata in P si indicele culorii cu care se vor colora toate detectiile gasite pentru sablonul curent (culoare). Calculam suma intensitatilor valorilor din sablon, dupa care calculam media acestora, si aplicam formula deviatei standard a intensitatilor grayscale pentru sablonul S. Apoi, pentru orice pozitii (x, y) pentru care sablonul S incape in imagine, calculam deviatia intensitatilor grayscale a pixelilor din fereastra determinata de elementul de pe "linia" x si "coloana" y de marimea sablonului, dupa care calculam corelatia dintre sablonul S si detectia(fereastra) curenta, iar daca este mai mare decat pragul dat ps, atunci realocam memorie pentru stocarea detectiei in tabloul fereastra, care retine pentru fiecare detectie indicele liniei x (coordonata inaltimii) , indicele liniei y (coordonata latimii), valoarea corelatiei dintre sablonul S si detectie si indicele culorii cu care se va colora chenarul detectiei.

9. Functia colorare_imagine are ca parametrii tabloul de pixeli P al imaginii folosite pentru template matching, latimea imaginii (latime_img) , latimea si inaltimea sablonului (latime_sablon si inaltime_sablon) folosite pentru a stii marimea ferestrelor al caror contur trebuie colorat. Parcurgem cele 2 linii si 2 coloane care reprezinta conturul ferestrei si atribuim pixelilor de pe aceste linii si coloane culoarea corespunzatoare cifrei a carei detectii o reprezinta.

10. Functia cmp_descrescator are ca parametrii 2 elemente de tipul void pentru care facem conversie in corpul functiei, aceasta este functia de compararare pentru qsort care compara valorile corelatiilor a 2 detectii.

11. Functia sortare_detectii are ca parametrii un tablou de detectii D alocat dinamic care contine elemente de tipul detectie(fiecare element contine indicele liniei si coloanei pe care se afla in imagine, corelatia care va fi criteriul dupa care vor fi sortate elementele acestui tablou si indicele culorii corespunzatoare cifrei a carei detectii o reprezinta) si numarul de detectii al tabloului D. Apelam functia qsort pentru tabloul D si folosim functia cmp_descrescator drept criteriu de comparare.

12. Functia eliminare_nonmaxime are ca parametrii tabloul de detectii D, numarul de detectii(nr_detectii), latimea si inaltimea sablonului (care sunt egale cu latimea si inaltimea unei detectii sau ferestre). parcurgem acest tablou D si eliminam toate detectiile care au suprapunerea spatiala mai mare decat 0.2 (eliminam detectia din dreapta a tabloului deoarece sunt sortate descrescator in functie de corelatie, deci in cazul in care cele doua detectii se intersecteaza, cea din stanga "castiga").

13. Functia main. (la inceput, am pus sub forma unui comentariu ordinea in care trebuie puse fisierele folosite in proiect). Salvam numele fiecarui fisier pe cate o linie a unui tablou bidimensional. Pentru modulul de criptare/decriptare, incarcam in memoria interna in tabloul P imaginea initiala folosita pentru criptare, ii salvam si latimea si inaltimea (latime_img si inaltime_img), dupa care o criptam cu functia criptare_BMP, o decriptam cu functia decriptare_BMP, afisam valorile testului chi square si

eliberam memoria pentru tabloul P. Pentru modulul de template matching, stocam intr-un tablou de tip pixel culorile pentru fiecare cifra, dupa care stocam imaginea pe care se face template matching in tabloul de pixeli P, transformam pixelii tabloului in grayscale, dupa care pentru fiecare cifra de la 0 la 9 incarcam pixelii sablonului cu cifra respectiva, latimea si inaltimea sablonului, transformam in grayscale pixelii acestuia si apelam functia template_matching obtinand astfel toate detectiile pentru cifra curenta, stocate pe cate o linie a tabloului bidimensional cu 10 linii (fereastra). Liniarizam si copiem acest tablou in noul tablou unidimensional alocat dinamic D, dupa care apelam functia de sortare a acestui tablou de detectii sortare_detectii si eliminam non-maximele cu functia eliminare_nonmaxime. Salvam din nou in tabloul de pixeli P imaginea initiala pe care am facut template matching si apelam functia de colorare a detectiilor colorare_imagine. Stocam headerul imaginii initiale, dupa care o rescriem, dupa rescriere continand si detectiile colorate. La sfarsit, dezalocam memoria folosita pentru tablouri.