

Proiect Programare Procedurala
Iliescu Andrei-Catalin, seria 13, grupa 133

Criptare-Decriptare

1. Structura *pixel* - defineste o structura pixel cu 3 campuri *unsigned char* pentru canalele de culoare red, green, blue.
2. Functia *xor* - returneaza xorul dintre doua numere de tip *unsigned char*
3. Functia *dimensiuneFisier* - returneaza dimensiunea unui fisier deschis
4. Functia *citire* - returneaza un pointer de tip *pixel* care pointeaza la inceputul unui vector alocat dinamic care retine sub forma liniarizata pixelii unei imagini date ca parametru. Functia citeste si retine octet cu octet datele din fisierul binar pe care le transforma apoi intr-o forma liniarizata pe pixeli.
5. Functia *scriere* - transforma forma liniarizata a unei imagini bitmap intr-un vector de octeti in ordinea in care acestia sunt retinuti de memoria calculatorului prin intermediul unui vector *liniar_recon* pe care il scrie intr-un fisier binar.
6. Functia *criptare* - afla inaltimea, latimea si paddingul imaginii transmise prin parametru. Citeste primul numar din fisierul care contine cheia secreta si cu ajutorul sau formeaza vectorul *r* cu numere random create cu ajutorul functiei *xorshift32*. Initializeaza un vector *tau* si apoi formeaza permutarea *tau* cu ajutorul vectorului *r*. Aplica permutarea *tau* pe vectorul de pixeli. Citeste din fisierul care contine cheia secreta al doilea numar si il transforma dintr-un numar de tip *unsigned int* in 4 numere de tip *char* pe care apoi le foloseste in cadrul formulei finale de criptare. Analog cu numerele din vectorul *r*. Functia foloseste apoi functia *scriere* pentru a salva noua imagine in memorie.
7. Functia *decriptare* - afla inaltimea, latimea si paddingul imaginii transmise prin parametru. Citeste primul numar din fisierul care contine cheia secreta si cu ajutorul sau formeaza vectorul *r* cu numere random create cu ajutorul functiei *xorshift32*. Initializeaza un vector *tau* si apoi formeaza permutarea *tau* cu ajutorul vectorului *r*. Formeaza apoi permutarea inversa *tau_invers*. Citeste din fisierul care contine cheia secreta al doilea numar si il transforma dintr-un numar de tip *unsigned int* in 4 numere de tip *char* pe care apoi le foloseste in cadrul formulei finale de decriptare. Analog cu numerele din vectorul *r*. Aplica permutarea *tau_invers* pe vectorul de pixeli. Functia foloseste apoi functia *scriere* pentru a salva noua imagine in memorie.
8. Functia *test_chi* - gaseste pentru fiecare *i*, $0 < i < 255$, frecventa de aparitie pentru fiecare canal de culoare r, g, b. Afla frecventa estimata pentru imaginea incarcata si apoi foloseste cele doua valori pentru a adauga numarul care reiese din formula data in suma aferenta fiecarui canal de culoare. Afiseaza apoi rezultatele obtinute.
9. Functia *main* - retine numele fisierelor cu care va lucra programul de criptare-decriptare si foloseste functiile de mai sus pentru prelucrarea imaginii date.