Nume: GRAURE MARIUS-ALEXANDRU

Grupa: 315CB

Readme Tema 3 - Programarea Calculatoarelor

Implementarea temei

Pentru implementarea acestei teme am urmat urmatorii pasi:

1. Citirea datelor de intrare din fisierul input.txt.

-> Se citeste linie cu linie si se extrag datele necesare: numele fisierului folosit pentru taskurile 1, 2 si 3, culoarea cu care vor fi inlocuite celelalte culori la taskul 1 (pentru memorarea acestei culori am folosit o structura de tip color ce contine 3 variabile de tip int ce reprezinta culoarea), cifrele ce vor fi eliminate in cadrul taskului 3 (pentru memorarea acestor numere am folosit un vector de intregi cu 10 elemente in care sunt salvate valori de tipul 0 daca cifra de pe pozitia i nu trebuie eliminata, 1 in caz contrar), si numele fisierului cu imaginea ce va fi folosita pentru bonus.

2. Citirea fisierului cu imaginea folosita la taskurile 1, 2 si 3

-> Dupa deschiderea fisierului pentru citire (fisier binar -> am folosit modul rb de deschidere) se citesc cele 2 headere in 2 structuri (una de tip bmp\_fileheader, si una de tip bmp\_infoheader, amandoua fiind definite in fisierul bmp\_header.h)

-> Dupa citirea celor 2 antete se salveaza offsetul curent pentru a cunoaste pozitia de inceput a matricii (folosit pentru determinarea paddingului). Apoi se citeste o matrice de width \* height (valori definite in antetul salvat in structura bmp\_infoheader). Aceasta matrice este de tip CColor (structura ce contine 3 variablie unsigned char). Se citesce din fisier un numar de octeti egal cu dimensiunea unei structuri CColor, iar tunci cand se ajunge la sfarsitul unei linii se face skip la padding folosind fseek.

3. Rezolvarea taskului 1

-> Pentru rezolvarea taskului 1 se parcurge o copie a matricei imaginii si fiecare pixel color, se modifica culoarea la valoarea celei citite din fisierul de input.

-> Imaginea este generata inserand initial cele 2 antete (acestea sunt deja setate de la citire). Apoi matricea de pixeli este scrisa, adaugand si paddingul.

4. Rezolvarea taskului 2

-> Pentru rezolvarea taskului 2 am declarat un vector cu 10 elemente (reprezentand cifrele) de matrici de dimensiune 7x7 (motivul pentru care am ales dimensiunea 7x7 si nu doar 5x5 este faptul ca am adaugat in jurul fiecarei cifre o margine de pixeli albi).

-> Se parcurge imaginea (intai pe coloane apoi pe linii) incepand cu un pixel imaginar la coordonatele (-1, -1) (pentru a putea include si marginea), si se construieste de fiecare data o matrice de 7x7 copiind elemntele de la pozitia curenta (i, j), si aceasta matrice este trimisa spre verificare unei functii

-> Functia compara matricea element cu element cu fiecare matrice reprezentand o cifra (cea despre care am mentionat anterior) iar in cazul in care cele 2 matrici se potrivesc, functia introarce ca rezultat cifra gasita, altfel se introarce -1 (considerat cod de eroare - nu a fost descoperita nici o cifra)

5. Rezolvarea taskului 3

In taskul anterior atunci cand descopar o cifra retin coordonatele (i, j) unde aceasta a fost descoperita.

In cadrul acestui task folosesc aceasta asociere astfel:

-> Parcurg cifrele in ordinea descoperirii lor (folosid i). Daca o cifra trebuie eliminata, sar peste indexul curent(i); Altfel copiez cifra de pe pozitia sa (i) pe pozitia la care a ajuns indexul (index); La final elimin toate cifrele ramase in imagine inlocuind pixelii colorati cu o culoarea alba (255, 255, 255);

6. Rezolvarea bonusului

Pentru rezolvarea bonusului am folosit acelasi principiu ca si la taskul 2. In plus am permis functiei de detectie a unei cifre sa ignore 1 culoare gresita.

7. Eliberarea memoriei

Am eliberat toata memoria alocata in cadrul programului.