Operatii posibile care se pot aplica asupra unei imagini:

• LOAD <fisier>

- Va încărca în memoria programului imaginea (PPM sau PGM) din fișierul dat.

- Dacă deja există un fișier încărcat în memorie, atunci acesta va fi înlocuit.

- Output-ul comenzii va fi:

  * Loaded <fisier> dacă operația s-a executat cu succes.

  * Failed to load <fisier> dacă operația a întâmpinat dificultăți.

- La încărcarea unui fișier nou, vom considera că acesta este automat selectat complet (întreaga suprafață).

- În cazul în care operația eșuează, programul va reveni la starea în care nu are niciun fișier încărcat.

• SELECT <x1> <y1> <x2> <y2>

- Restrânge efectul comenzilor următoare la pixelii incluși în intervalul [x1, x2) pe axa X (lățimea imaginii), respectiv în intervalul [y1, y2) pe axa Y (înălțimea imaginii).

∗ Prin SELECT <x1> <y1> <x2> <y2>, se vor selecta pixelii [x1, x2), [y1, y2). Spre exemplu,

SELECT 0 0 1 1 va selecta doar pixelul (0, 0); în vreme ce SELECT 0 0 2 2 va selecta pixelii

(0, 0), (0, 1), (1, 0), (1, 1).

- Originea axelor de coordonate se află în colțul din stânga sus.

- Output-ul comenzii va fi:

  * Selected <x1> <y1> <x2> <y2> dacă operația s-a executat cu succes.

  * Invalid coordinates dacă vreunul din punctele de delimitare se află în afara imaginii.

  * No image loaded dacă nu există un fișier încărcat.

- După o selecție invalidă, se va păstra selecția anterioară.

- Ordinea (x1 < x2 și y1 < y2) nu este garantată. Va trebui tratate ambele cazuri.

• SELECT ALL

- Readuce zona de selecție la întreaga imagine.

- Output-ul comenzii va fi:

  * Selected ALL dacă operația s-a executat cu succes.

  * No image loaded dacă nu există un fișier încărcat.

• HISTOGRAM <x> <y>

- Va afișa histograma imaginii folosind maxim x stele și y bin-uri.

- Valorile permise pentru y sunt puterile lui 2 din intervalul [2, 256].

- Output-ul comenzii va fi:

  * Histograma, dacă aceasta se poate calcula.

  * Black and white image needed dacă imaginea este color.

  * No image loaded dacă nu există un fișier încărcat.

• EQUALIZE

- Va face egalizarea imaginii alb-negru conform metodei descrise în secțiunea aferentă a temei.

- Output

-ul comenzii va fi:

  * Equalize done dacă operația s-a efectuat cu succes.

  * Black and white image needed dacă imaginea este color.

  * No image loaded dacă nu există un fișier încărcat.

- Operația se va realiza mereu pe toată imaginea, indiferent de selecția curentă. Deși operația se va aplica asupra întregii imagini, nu se va realiza niciun fel de SELECT (și, în particular, niciun SELECT ALL), astfel că selecția curentă a imaginii nu va fi modificată.

• ROTATE <unghi> (bonus)

- Va face rotirea selecției curente cu un anumit număr (exprimat în grade).

- Valorile permise pentru unghi sunt +/-90, +/-180, +/-270, +/-360.

- În cazul în care valoarea unghiului este pozitivă, rotirea selecției se va face spre dreapta. Dacă valoarea unghiului este negativă, rotirea se va face spre stânga.

- Output-ul comenzii va fi:

  * Rotated <unghi> dacă operația s-a efectuat cu succes.

  * The selection must be square dacă selecția nu este compatibilă cu cerința.

  * Unsupported rotation angle dacă valoarea unghiului de rotație nu este unul dintre cele permise.

  * No image loaded dacă nu există vreun fișier încărcat.

- Observație: ROTATE se poate aplica doar pentru o selecție de dimensiune NxN (submatrice pătratică) sau pe întreaga imagine.

• CROP

- Va reduce imaginea la cea cuprinsă în cadrul selecției curente.

- După CROP, orice comandă, inclusiv SELECT ALL, va face referire doar la subimaginea rezultată în urma comenzii CROP.

- Când operația se execută cu succes, comanda va avea drept output mesajul: Image cropped.

- Dacă înainte de executarea comenzii nu se încarcă nicio imagine, output-ul comenzii va fi No image loaded.

• APPLY <PARAMETRU>

- Disponibilă doar pentru tipurile de imagini color. Această comandă va aplica peste selecție unul din nucleele de imagine prezentate în secțiunea "Formatele de imagini", în funcție de parametrul introdus:

  * <EDGE>

  * <SHARPEN>

  * <BLUR>

  * <GAUSSIAN_BLUR>

- Modul de aplicare a transformării este următorul:

  * Se înmulțesc valorile pixelilor din imagine cu valorile corespunzătoare ale pixelilor din kernel.

  * Mai apoi, rezultatul noului pixel este suma elementelor menționate.

  * Matricea kernel se va aplica pe toate cele 3 canale ale imaginii color.

- Output-ul comenzii va fi “APPLY <PARAMETRU> done”.

- În cazul în care nu se introduce parametrul corespunzător, se va afișa “AP

PLY parameter invalid”.

- În cazul în care APPLY se aplică asupra imaginilor grayscale, se va afișa "Easy, Charlie Chaplin".

- Dacă înainte de executarea comenzii nu se încarcă nicio imagine, output-ul comenzii va fi No image loaded.

- Puteți găsi o explicație detaliată despre cum funcționează aceste nuclee aici.

- Pixelii care nu au destui vecini pentru a putea calcula noile valori vor rămâne nemodificați.

- Observație: Prin aplicarea filtrului EDGE se pot obține valori în afara intervalului [0, 255]. Pentru tratarea acestui caz, se va aplica funcția clamp. Astfel, valorile negative vor deveni 0, iar cele care depășesc 255 vor fi limitate la această valoare.

• SAVE <nume_fisier> [ascii]

- Salvează imaginea curentă în fișierul nume_fisier.

- Dacă parametrul ascii este prezent, atunci valorile pixelilor vor fi salvate în format ASCII. Altfel, vor fi salvate în format binar.

- Formatul (binary/text) în care se va salva imaginea este specificat la apelul comenzii SAVE. Acest format trebuie respectat ca atare, indiferent de formatul pe care l-a avut imaginea încărcată inițial.

- În urma operației SAVE, ultima imagine încărcată prin operația LOAD va rămâne încărcată în memorie.

- Dacă încă nu s-a încărcat o imagine, atunci se va afișa mesajul "No image loaded."

- Comanda va afișa, după salvarea fișierului, mesajul Saved <nume_fisier>.

• EXIT

- Face închiderea graceful a programului.

- Acest proces înseamnă dealocarea tuturor resurselor, închiderea fișierelor și oprirea execuției programului.

- Nu se poate realiza fără a avea o imagine încărcată în prealabil.

- În acest caz, output-ul comenzii va fi "No image loaded".

O linie care nu poate fi interpretată ca una dintre comenzile de mai sus va avea ca rezultat afișarea mesajului "Invalid command" și nu va genera alte efecte.

Mod implementare problema:

In main, am declarat toate variabilele de care voi avea nevoie in rezolvarea problemei. De asemenea, am plasat apelurilor functiilor ce rezolva toate operatiile care pot fi aplicate in cadrul temei. In cadrul unei instructiuni repetitive, am decis sa citesc cate un rand, pe care sa il separ in cuvinte cu ajutorul functiei word_to_word si a unei matrici de caractere.

Operatia "LOAD" necesita ca inainte de a fi apelata functia, in cazul in care mai fusese incarcata in memorie o alta fotografie, sa dealoce elementele matricii pixelilor. In functia LOAD, daca fisierul putea fi deschis, dupa deschiderea acestuia, citeam din fisier in modul read tipul imaginii (ex: P2, P3, P5), si inca un string pe care l-am denumit line. De acest vector, ma voi folosi pentru a ignora comentariile si a retine numarul coloanelor, al liniilor si al limitei superioare a valorilor pixelilor. In cazul in care tipul matricii era P2 sau P3, insemna ca citirea din fisier a elementelor matricii se puteau face cu un simplu fscanf, precedat de alocarea dinamica a matricii conform dimensiunilor specifice. In caz contrat, a fost nevoie sa retin unde ma aflu in fisier dupa ce am citit limita superioara, sa il inchid, sa il deschid iar in modul read binary si sa ma duc la octetul la care ramasesem initial in fisier.

Operatia "SELECT" si "SELECT ALL" au fost realizate in aceeasi functie. In cazul in care nu trebuia selectata toata matricea, m-am folosit de functia atoi si am verificat daca selectia se poate realiza cu adevarat, daca respecta toate conditiile pentru a fi valida.

Operatia "HISTOGRAM", in cazul in care parametrii erau introdusi in mod corect, iar imaginea era una alb-negru, m-am folosit de functia atoi. Am calculat frecventa fiecarui pixel din imagine. Dupa, calculez restul elementelor necesare pentru aplicarea formulei date.

Operatia "EQUALIZE", in cazul in care imaginea era de tip alb-negru, am calculat frecventa fieecarui pixel, aria sectiunii si suma frecventelor pentru k de la 0 la valoarea pixelului. M-am folosit de functia clamp pentru a incadra valoarea introdusa in functie in intervalul [0, 255].

Operatia "CROP" se va realiza diferit in functie de tipul imaginii incarcate. Aloc o functie auxiliara, in care copiez elementele matricii ce fac parte din sectiunea aleasa. Dupa care, dealoc matricea mare, pentru a o aloca cu noile ei dimensiuni si copiez elementele din cea auxiliara in matricea nou alocata. Dealoc auxiliarele.

Operatia "APPLY" s-a realizat dupa testarea validatii comenzii, a incarcarii in memorie a unei imagini si a faptului daca imaginea este de tip color sau nu. Pentru fiecare dintre parametrii dati in enuntul problemei, am adaugat matricile de dimensiune 3 x 3, intr-una auxiliara de care m-am folosit la mod general. Am realizat o copie a matricii si am calculat care sunt sectiunile de matrice pe care voi lucra. Dupa care, am apelat functia solve_pixel in care am insumat produsele elementelor matricii de 3 x 3 careia elementul calculat ii reprezinta kernel, cu elementele matricii auxiliare. Dupa care am copiat din copia matricii, in cea initiala si m-am folosit iarasi de functia clamp pentru a pastra toate elementele in intervalul [0, 255].

Operatia "SAVE" s-a realizat in modul urmator: am deschis fisierul in modul write pentru a adauga tipul imaginii, numarul de coloane, de linii si limita superioara, lucru pe care l-am facut atat pentru salvarea in mod ascii, cat si in mod binar. La salvarea ascii am continuat prin a vedea daca imaginea este de tip color, pentru a sti cate elemente am de introdus in fisier, lucru realizat cu fprintf. In schimb, la salvarea in mod binar, am continuat prin a inchide fisierul si a-l deschide in modul append binary pentru a relua scrierea de unde am ramas in fisier, in mod binar. De aceasta data, introducerea elementelor matricii in fisier s-a realizat cu fwrite si s-a facut element cu element.

Operatia "ROTATE" s-a rezumat la aplicarea functiei rotate_90 de mai multe ori. Aceasta a fost facuta la modul general, pentru a se putea aplica si pentru o submatrice de dimensiuni n x n, dar si pentru momentul in care trebuie rotita toata matricea. De asemenea, procedeul aplicat a fost diferit in functie de tipul matricii. Pentru imaginile P2 si P5 am alocat 2 matrici auxiliare. In prima auxiliara am copiat submatricea care trebuie intoarsa cu 90 de grade la dreapta, iar in cea de-a doua auxiliara s-a produs efectiv rotatia. In cazul in care toata matricea trebuia intoarsa, am dealocat matricea mare si am alocat-o astfel incat sa pot realiza copierea auxiliarei 2 in aceasta fara probleme. Pentru imaginile P3 si P6 procedeul este asemenator, dar difera modul in care matricea este intoarsa.

Concepte din curs care au fost acoperite:

- Impartirea in fisiere
- Realizarea unui Makefile nou
- Adaugarea elementelor de programare defensive
- Adaugarea de Macro-uri