

Esame di Programmazione Cl. B (Parte 2)

27 Gennaio 2022

Ingegneria e Scienze Informatiche A.A. 2021-2022

Si vuole realizzare una soluzione che implementi alcune funzionalità di salvataggio e trasformazione di immagini.

Queste immagini sono gestite come una **matrice di pixel**.

La struttura di un'immagine è infatti la seguente:

```
struct image {  
    struct pixel** pixels;  
    int width;  
    int height;  
};
```

Un **pixel** viene descritto mediante tre valori: **R**, **G** e **B**, che vanno da **0** a **255** e rappresentano l'intensità dei tre colori primari Red, Green e Blue.

La struttura di un pixel è la seguente:

```
struct pixel {  
    unsigned char R;  
    unsigned char G;  
    unsigned char B;  
};
```

Lo 0 è l'intensità minima, pertanto RGB(0,0,0) rappresenta il colore **nero**, mentre il 255 è l'intensità massima, quindi RGB(255,255,255) rappresenta il colore **bianco**.

Si deve consegnare un file **image_manager.c** contenente l'implementazione delle funzioni contenute nel file header **image_manager.h**.

Le funzioni da implementare sono le seguenti:

- **RGB_to_greyscale (5 punti)**: ritorna un'immagine in scala di grigi ottenuta dall'elaborazione dell'immagine presa in input. I tre valori R, G e B di ciascun pixel della nuova immagine devono avere la medesima intensità, ottenuta dalla media dei valori R, G e B del pixel originale.

La signature della funzione è la seguente:

```
struct image RGB_to_greyscale(struct image img);
```

- **add_border (5 punti):** prende in input un'immagine e un colore. Ritorna l'immagine con l'aggiunta di un bordo spesso **1 pixel** del colore indicato.

La signature della funzione è:

```
struct image add_border(struct image img, struct pixel color);
```

- **load_image (3 punti):** prende in input un file contenente un'immagine, lo legge e restituisce l'immagine caricata. Se l'operazione non va a buon fine restituisce un'immagine **vuota** (height=0, width=0, pixels = NULL).

La signature della funzione è la seguente:

```
struct image load_image(char* filename);
```

- **save_image (3 punti):** prende in input un'immagine e la salva su un file. Se l'operazione va a buon fine ritorna **1**, altrimenti ritorna **0**.

La signature della funzione è la seguente:

```
int save_image(struct image img, char* filename);
```

I file che immagazzinano le immagini devono avere la seguente formattazione:

- La **prima riga** deve contenere due numeri interi positivi: il primo rappresenta l'**altezza** dell'immagine (numero di righe), il secondo la **larghezza** (numero di colonne).
- Ciascuna **riga** del file, successiva alla prima, contiene **un pixel**, rappresentato da una tripletta di valori (R, G e B)
- I valori numerici sono separati da un carattere di **tabulazione**

Esempio:

Questo file rappresenta un'immagine 4×3 (HxW) con:

- un punto in alto a sinistra bianco e il resto della prima riga nero.
- una riga grigia
- due righe con punti rossi (255, 0, 0), verdi (0, 255, 0), blu (0, 0, 255), azzurri (0, 255, 255), gialli (255, 255, 0) e magenta (255, 0, 255).

```
4 3
255 255 255
0 0 0
0 0 0
127 127 127
127 127 127
127 127 127
255 0 0
0 255 0
0 255 255
0 0 255
255 255 0
255 0 255
```

Esempio 2:

Questo file rappresenta la stessa immagine del file mostrato sopra, trasformata in un'immagine in scala di grigi e con l'aggiunta di un bordo bianco spesso 1 pixel:

```
6 5
255 255 255
255 255 255
255 255 255
255 255 255
255 255 255
255 255 255
255 255 255
0 0 0
0 0 0
255 255 255
255 255 255
127 127 127
127 127 127
127 127 127
255 255 255
255 255 255
85 85 85
85 85 85
170 170 170
255 255 255
255 255 255
85 85 85
170 170 170
170 170 170
255 255 255
255 255 255
255 255 255
255 255 255
255 255 255
255 255 255
```