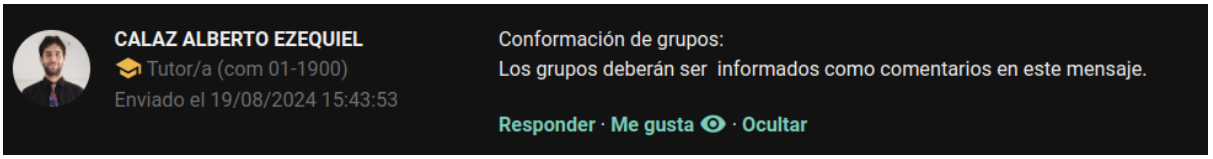
Trabajo Práctico

Manipulación de imágenes sin compresión

Modalidad de entrega

La entrega es grupal, en grupos de 3 personas.
En caso de conformarse todos los grupos necesarios y que aún haya estudiantes sin grupo, estos serán asignados por los docentes a grupos previamente armados, formando así grupos de 4 personas.

Los grupos deben ser formados y notificados vía foro de MIEL hasta las 23:59 del domingo 25 de agosto de 2024 como comentarios en la siguiente publicación:



El nombre de cada grupo debe ser una palabra, y no se puede repetir con otro grupo. El nombre del grupo debe ser una palabra que figure en el diccionario de la RAE, que pueden consultar en el siguiente link: https://dle.rae.es/

El nombre del grupo debe estar formado por letras cuyo valor ASCII esté comprendido entre 0x41 y 0x5a (inclusive). Eso implica que no se admiten espacios, letras en minúscula, tildes, números, etc.

Ejemplos de nombres que serán rechazados:

Palabra	Motivo de rechazo
LOS PIOJOS	Tiene un espacio (0x20)
La Renga	Tiene un espacio y letras en minúsculas
ASDF	Aviso: La palabra asdf no está en el Diccionario
C++	El ASCII de ‘+’ es 0x2b, no está en el rango solicitado

Ejemplo de nombre válido:

Palabra	Motivo de aceptación
INVISIBLE	Está formado por caracteres válidos, y según la RAE su definición es: 1. adj. Que no puede ser visto.

Se deberá entregar un archivo con el siguiente formato:
TP_TOPICOS_2024_2C_LUNES_{NOMBRE_DE_GRUPO}.zip
Por ejemplo: Si el grupo se llamase “INVISIBLE” -y sus integrantes fueran Spinetta, Pomo y

Machi- el archivo debería llamarse TP_TOPICOS_2024_2C_LUNES_INVISIBLE.zip
El formato de entrega es motivo de rechazo del TP.
Como ocurre con cualquier sistema, debe respetar el formato solicitado.
Ejemplos de archivos que se considerarán incorrectos:

Nombre de archivo	Motivo de rechazo
TP_TOPICOS_2024_2c_LUNES_INVISIBLE.zip	contiene una c minúscula en el nombre
TP_TOPICOS_2024_2C_LUNES_INVISIBLE(1).zip	contiene (1) en su nombre
TP_TOPICOS_2024_2C_MARTES_INVISIBLE.zip	día incorrecto
TP_TOPICOS_2024_2C_LUNES_INVISIBLE.rar	formato de archivo incorrecto

Dentro de ese archivo debe haber un archivo llamado “funciones_grupo.h” (sin importar el nombre del grupo), que invoque a los dos archivos de cada estudiante.

Por ejemplo, funciones_grupo.h debe contener los includes necesarios para los archivos de Luis Alberto Spinetta, Pomo Lorenzo y Machi Rufino.
Quedando así:

```
#include “funciones_spinetta.h”
#include “funciones_pomo.h”
#include “funciones_rufino.h”
```

De modo que la entrega total quedará con los siguientes archivos:
`funciones_grupo.c`
`funciones_grupo.h`
`funciones_spinetta.c`
`funciones_spinetta.h`
`funciones_pomo.c`
`funciones_pomo.h`
`funciones_machi.c`
`funciones_machi.h`

Los apellidos serán en minúsculas. Si algún apellido tuviera tildes, espacios, o cualquier carácter ASCII que no esté comprendido entre 0x41 y 0x5a, éste deberá ser modificado a un formato simplificado reemplazando estos caracteres por algunos válidos, u omitiéndoles en caso de los espacios.

Por ejemplo:
Carlos Alberto García debería entregar archivos llamados
`funciones_garcia.c y funciones_garcia.h`

Mientras que Juan Del Barrio debería entregar
`funciones_delbarrio.c y funciones_delbarrio.h`

El archivo funciones_grupo.c debe contener un comentario al comienzo con información sobre sus integrantes en el siguiente formato:
Cada grupo debe completar los datos de sus integrantes en el siguiente formato:
Apellido(s), nombre(s):
DNI: (sólo números, sin puntos)
Entrega: Sí/No.

Ejemplo válido:	Ejemplo inválido:
Apellido: Spinetta, Luis Alberto *DNI: 12345678* *Entrega: Sí*	*Apellido: Luis Alberto Spinetta* *DNI: 12.345.678* *Entrega: Sí*

El ejemplo inválido contiene dos errores:

- No respeta el formato “apellido(s), nombre(s)”
- Utiliza puntos en el número de DNI

El campo “Entrega:” es para indicar si el/la estudiante hace entrega del TP. Podría algún integrante del grupo abandonar la cursada y no hacer entrega. En ese espacio pueden notificar el motivo, o cualquier dato que consideren oportuno.

Del mismo modo que el nombre del archivo es motivo de rechazo del TP, el contenido del mismo también lo es.
Si algún integrante no tuviera completos sus datos en funciones_grupo.h, la entrega -grupal- será inválida.

Si el archivo entregado contuviera algún archivo/carpeta distinto a lo solicitado, la entrega será rechazada.

Será responsabilidad de todos los integrantes del grupo verificar que los datos de los/las integrantes sean correctos, y que los archivos estén en el formato solicitado (destacando nuevamente que no se aceptarán archivos que contengan carpetas en su interior).

El trabajo práctico tiene valor de parcial y consta de una entrega (grupal) y una defensa (individual).

Recuerden que cualquier consulta sobre el formato de la entrega, o cualquier asunto relativo al TP, deberá hacerse vía foro de miel.

Se recomienda enfáticamente no esperar a último momento para resolver el TP. De ese modo tendrán más tiempo para resolver cada problemática que pueda surgir con el mismo.

Modalidad de defensa

La defensa se hará el lunes 28 de octubre, en laboratorio. Se solicitará hacer modificaciones en el programa para extender sus funcionalidades.

Requisitos de entrega

La resolución del TP debe ser entregada como práctica vía plataforma MIEL. La fecha de vencimiento de la entrega es el día domingo 15 de septiembre a las 23:59 hs.
Las entregas que no respeten la fecha indicada no serán válidas.

El trabajo práctico entregado debe funcionar correctamente, sin warnings y cumplir con todas las funcionalidades requeridas.
Deberá cumplir con las especificaciones de nombre y formato de archivo, de lo contrario la entrega será inválida y el TP desaprobado.
Los grupos deberán revisar adecuadamente el trabajo práctico previo a la entrega, dado que una vez entregado no se aceptarán re entregas.

Una vez entregado, en caso de cumplir con las condiciones planteadas de fecha, resolver adecuadamente cada uno de los puntos indicados, no tener warnings y tener el formato adecuado de archivo, el TP pasará a estar en estado “entregado”.

En caso de no cumplir con alguno de los requisitos, se podrá utilizar la fecha de recuperatorio (quitando por esto la posibilidad de recuperar el parcial) para hacer una nueva entrega, y defensa. La nueva entrega tendrá nuevos requerimientos que se informarán debidamente. Y la defensa tendrá las mismas condiciones que la defensa inicial.

Enunciado

Se solicita realizar un programa en lenguaje C llamado *bmpmanipuleitor*, que a partir de un archivo BMP de 24 bits de profundidad genere otro archivo BMP con las modificaciones solicitadas como argumentos a main.

Por ejemplo:

```
bmpmanipuleitor.exe --negativo imagenOriginal.bmp
```

Debe generar un archivo llamado *GRUPO_negativo_imagenOriginal.bmp*
que contenga la misma imagen, pero con los colores invertidos.

De modo que si el grupo INVISIBLE quisiera aplicar este mismo filtro a una imagen llamada unlam.bmp, la imagen resultante debería llamarse
INVISIBLE_negativo*_unlam.bmp*

Del mismo modo, deberá soportar las siguientes funcionalidades:
--escala-de-grises // deberá promediar los valores de cada color RGB y transformarlo a gris

`--espejar-horizontal` // deberá invertir horizontalmente la imagen
`--espejar-vertical` // deberá invertir la imagen verticalmente
`--`aumentar-contraste=10 // aumenta el contraste en un 10%
`--`reducir-contraste=20 //reduce el contraste en un 20%
`--`tonalidad-azul=50 // aumenta en un 50% la intensidad del color azul
`--`tonalidad-verde=5 // aumenta en un 5% la intensidad del color verde
`--`tonalidad-roja=15 // aumenta en un 15% la intensidad del color rojo
`--`recortar=30 // reduce el tamaño de la imagen al 30%, sin cambiar sus proporciones, solamente descarta lo que exceda ese tamaño. Por ejemplo: una imagen de 1000px x 500px, deberá mantener todos los píxeles que estén entre 0 y 299 en el eje X y entre 0 y 149 el eje Y.

`--`achicar=10 // reduce la imagen al 10%. Por ejemplo: una imagen de 1000px x 500px, deberá reescribir todos los píxeles para que se ubiquen entre 0 y 99 en el eje X y entre 0 y 49 el eje Y, formando la misma imagen inicial (con peor calidad, claro).

Estas últimas siete funcionalidades requieren que el porcentaje llegue como argumento y tolere cualquier número entre 0 y 100 (no es necesario que pueda aumentar un 200%, por decir, deben validar que los parámetros tengan el rango adecuado).

`--`rotar-derecha // gira la imagen 90 grados a la derecha (sentido horario)
`--`rotar-izquierda // gira la imagen 90 grados a la izquierda (sentido antihorario)

`--`concatenar-horizontal // recibirá como argumento dos imágenes, y creará una nueva con el contenido de ambas, una al lado de la otra. En caso de tener distinto alto, la más pequeña se rellenará con algún color a gusto (que no sea ni blanco ni negro, dejando de lado el debate de que sean o no un color) hasta completar el alto.

`--`concatenar-vertical // recibirá como argumento dos imágenes, y creará una nueva con el contenido de ambas, una arriba de la otra, en caso de tener distinto ancho, la más pequeña se rellenará con algún color a gusto (que no sea ni blanco ni negro, dejando de lado el debate de que sean o no un color) hasta completar el ancho de la otra

`--`comodin // una funcionalidad nueva, decidida por el grupo (no se puede repetir con alguna de las solicitadas previamente en el TP)

Nota: los argumentos pueden enviarse de a uno, o de a varios, de modo que en un solo llamado al programa, ejecute todas las modificaciones y genere todos los archivos necesarios. Si un parámetro es incorrecto, el error debe ser informado por consola, y ejecutar todos los otros filtros.

Si algún comando no pudiese ejecutarse, debe informarse el motivo (por ejemplo: no se pueden concatenar las imágenes porque sólo se haya enviado una por argumentos, archivo no encontrado, etc).

Un llamado válido al programa podría ser:

bmpmanipuleitor.exe --negativo unlam.bmp --escala-de-grises --argumento-incorrecto-que-debera-ser-ignorado --aumentar-contraste=18 --negativo

Ese llamado debería generar los archivos

- *INVISIBLE_escala-de-grises_unlam.bmp*
- *INVISIBLE_negativo_unlam.bmp*
- *INVISIBLE_aumentar-contraste_unlam.bmp*

El argumento *--negativo* aparece dos veces, deberá ejecutarse sólo una. El argumento incorrecto no deberá generar ningún archivo ni abordar la ejecución del programa. Pero sí informar el error en pantalla.

Todas las modificaciones deben hacerse sobre los archivos previamente mencionados, y no en el main.

Los archivos deben quedar ubicados en la misma ubicación desde donde se haya llamado al binario; no serán aprobados los TP que guarden imágenes en alguna carpeta hardcodeada.

Algunos puntos requieren modificaciones en el encabezado del archivo BMP, cuya descripción es la siguiente:

Bytes	Información
0, 1	Tipo de fichero "BM"
2, 3, 4, 5	Tamaño del archivo
6, 7	Reservado
8, 9	Reservado
10, 11, 12, 13	Inicio de los datos de la imagen
14, 15, 16, 17	Tamaño de la cabecera del bitmap
18, 19, 20, 21	Anchura (píxels)
22, 23, 24, 25	Altura (píxels)
26, 27	Número de planos
28, 29	Tamaño de cada punto
30, 31, 32, 33	Compresión (0=no comprimido)
34, 35, 36, 37	Tamaño de la imagen
38, 39, 40, 41	Resolución horizontal
42, 43, 44, 45	Resolución vertical
46, 47, 48, 49	Tamaño de la tabla de color
50, 51, 52, 53	Contador de colores importantes

Muchos campos de este encabezado no son necesarios para la realización del TP, sí es necesario contemplar que hay campos de 16 y de 32 bits y que deben ser almacenados en el tipo de dato apropiado, para su correcta interpretación.
Alto, ancho, tamaño de imagen, tamaño de cada punto/píxel, inicio de los datos de la imagen son datos de suma importancia para la realización del TP.

Detalles a tener en cuenta

Lo buscado en este TP es que para cada filtro se utilice una matriz: es opcional que ésta sea dinámica o estática.
No se aceptarán entregas que trabajen sobre un array unidimensional, ni tampoco que trabajen directamente sobre el archivo, utilizando fseek para posicionarse en distintos lugares del mismo. Uno de los principales objetivos de este TP es que aprendan a trabajar con matrices.

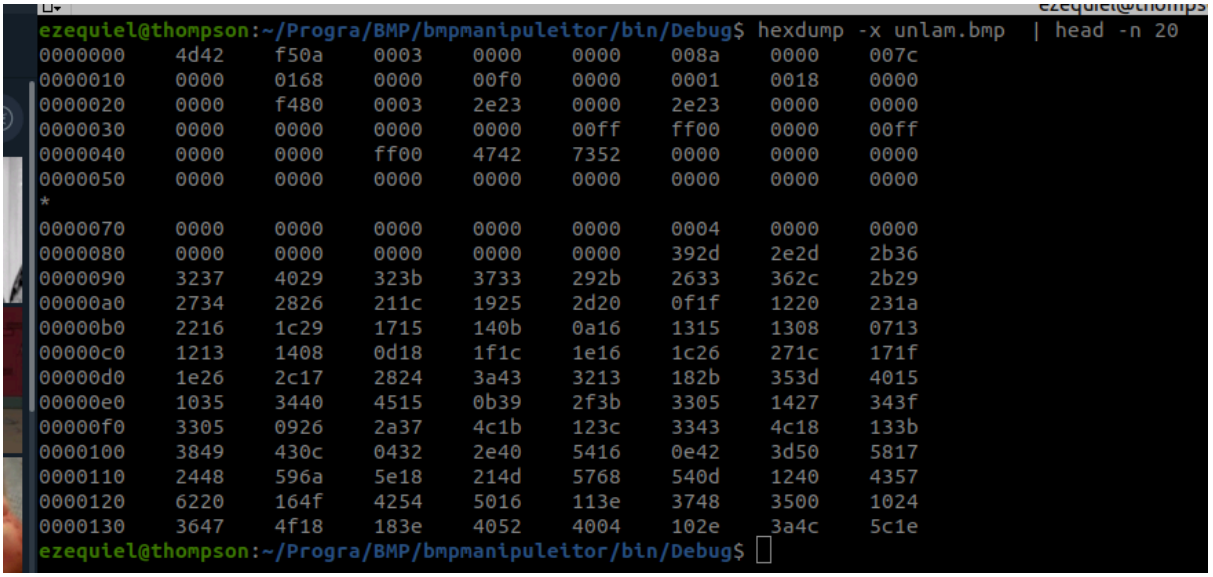
Es posible que al trabajar con imágenes grandes el programa no funcione adecuadamente porque el tamaño de las matrices supere al tamaño del área stack. Eso no será un problema, para las correcciones utilizaremos imágenes pequeñas.

Se valorará positivamente que utilicen memoria dinámica (pudiendo así trabajar con imágenes grandes), que trabajen adecuadamente con el padding de los archivos (para entender este concepto, sugiero utilizar imágenes que tengan un tamaño que no sea múltiplo de 8, por ejemplo una imagen de 239 píxeles de ancho), o cualquier mejor que quieran hacer.

Detalles a tener en cuenta

Una imagen como las que vamos a utilizar en este TP está formada por un encabezado y un array de bits que representa la imagen. Este array de bits, al haber definido que cada pixel tiene 24 bits de profundidad, se debe interpretar como un píxel al contenido de 3 bytes consecutivos.

Podría ser de utilidad utilizar un programa que haga un dump hexadecimal, si no conocen ninguno… podrían hasta resolverlo como punto extra.



Los tres bytes consecutivos son valores que van entre 0 y 255 (unsigned char), o más claramente en hexadecimal: 0x00 hasta 0xff, que indican qué cantidad de color (rojo, verde o azul) va a tener la imagen.
Si las tres variables BRG (Blue, Red y Green; o en español Azul, Rojo y Verde) están igualadas, el color será gris, y dependiendo de su intensidad, se acercará más al blanco o al negro.
El color 0xffffff equivale al blanco, mientras que el 0x000000 equivale al negro.
Si quisiéramos escribir un color verde puro, debería ser 0x0000ff (porque tiene cero en la componente roja, cero en la componente azul y 255 en la componente verde).
De ese modo se pueden armar 2^24 colores distintos.