Yeditepe University
Department of Computer Engineering

CSE 232 Systems Programming
Fall 2022

Term Project

Image Editor

Due to: 2 January 2023

In this project, you will implement a simple **image editor** named `Iedit`.

Write your image editor in C using gfx graphics tool. Use gcc compiler on Linux.

The editor must perform the following functions:
- E (edit): Opens the specified bitmap file and a window on the screen. If the file exists, its contents are drawn on the window.
- F (filter): Applies the following operations on the image:
  - cp - copy/paste
  - bl - blur
  - gr - grayscale
  - br - brighten
- U (undo): Discards the last operation.
- R (redo): Repeats the last operation.
- V (view): Views the image on the window.
- S (save): Saves the file
- X (exit)

The editor must keep the original image in a memory buffer and edit it in an editing buffer to display. Use the following data structures:

```
struct pixel {
       int    red;
       int    green;
       int    blue;
}
struct pixel orig[128][128]; // original image (max. image size 128x128)
struct pixel eb[128][128];   // editing buffer
```

For undo/redo, you need to keep all operations performed on the image. Use the following stack structure:

```
struct operation {
       int    op;  // 1-copy/paste, 2-blur, 3-grayscale, 4-brighten
       int    x;   // destination for copy/paste
       int    y;
}
struct operation stack[100];     // max. 100 operations
struct operation *sptr;          // stack pointer
struct operation *max_sptr;      // maximum stack pointer for redo
```

Write the following functions to implement the editor commands:

`edit(char *filename)`
Opens the specified bitmap file, reads its contents into the oriinal image `orig[][]` and the editing buffer `eb[][]`. If the file does not exist, it gives an error message. Then, it must open a window on the screen, copy the image to the viewing buffer and draw it on the screen. File name must be given by the user in the argument list when he/she runs `Iedit`.

`filter(int op, int x, int y)`
Pushes the operation on the stack.

`undo()`
Updates the stack pointer. (Check if the stack is empty)

`redo()`
Updates the stack pointer. (Check if the stack pointer exceeds the valid operations on the stack)

`view()`
Applies all filters, that are stored on the stack, to the image in the editing buffer. Then, it displays the contents of the editing buffer on the screen.

`save(char *filename)`
Applies all filters, that are stored on the stack, to the image in the editing buffer, then writes the editing buffer to the file in the correct format and closes the file.

Write the following functions to implement editor operations:

`copy_paste(int x, int y)`
Copies the top left part of the image from (0,0) to (20,20) and pastes it starting from (x,y).

`blur()`
Blurs the image by setting the red, green, blue components of each pixel value to the average of its 4 neighbour pixels:

$x_{i,j} = (x_{i,j} + x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j=1})/5$

`grayscale()`
Converts the image to grayscale by setting each pixel value to the average of its red, green, blue components:

$gray = (xred_{i,j} + xgreen_{i,j} + xblue_{i,j})/3$
$xred_{i,j} = gray$
$xred_{i,j} = gray$
$xred_{i,j} = gray$

`brighten()`
Changes the brightness of the image by incrementing the red, green, blue components of each pixel:

$xred_{i,j} = xred_{i,j} + brightness;$ if ($xred_{i,j} > 255$) $xred_{i,j} = 255;$ if ($xred_{i,j} < 0$) $xred_{i,j} = 0;$
$xgreen_{i,j} = xgreen_{i,j} + brightness;$ if ($xred_{i,j} > 255$) $xred_{i,j} = 255;$ if ($xred_{i,j} < 0$) $xred_{i,j} = 0;$
$xblue_{i,j} = xblue_{i,j} + brightness;$ if ($xred_{i,j} > 255$) $xred_{i,j} = 255;$ if ($xred_{i,j} < 0$) $xred_{i,j} = 0;$

Implement the editor using the following algorithm:

```
call edit()
read input from keyboard and parse it
while (input is not X) {
       if input is F then call filter()
       if input is U then call undo()
       if input is R then call redo()
       if input is V then call view()
       if input is S then call save()
       read input from keyboard and parse it
}
exit
```

You can download gfx from:
https://www3.nd.edu/~dthain/courses/cse20211/fall2013/gfx/

Example:

Editor commands:

```
./Iedit  myimage.bmp
F bl          → pushes blur operation on the stack
F gr          → pushes grayscale operation on the stack
U             → updates stack pointer
V             → applies operations on the stack to the image and displays
R             → updates stack pointer
V             → applies operations on the stack to the image and displays
R             → error
F cp 50 50    → pushes copy operation on the stack (copy (0,0)-(20,20) to (50,50)-(70,70))
S
X
```