Goal:

Manipulation of BMP image files:
(1) Reducing the size of an image by a factor of 2.
(2) Increasing the size of an image by a factor of 2.

Objective: Developing experience with C arrays and pointers.

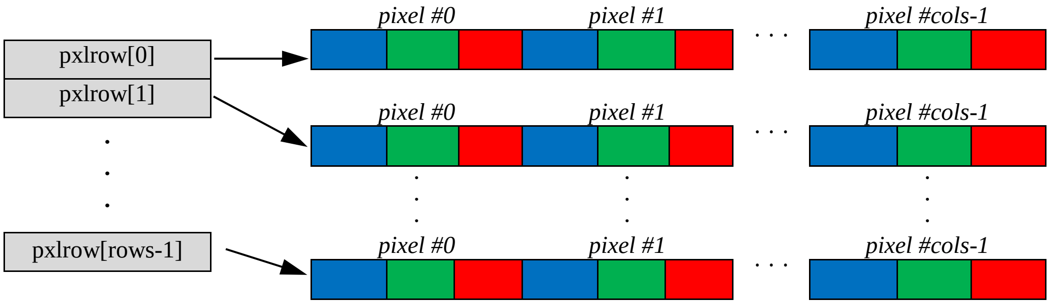
Background: When the size of a digital image is either reduced or enlarged, the pixels that form the image become increasingly visible, making the image appear soft if pixels are averaged, or jagged if not. Although algorithms exist to reduce the effect, this assignment is only to resize the image.

Download: Download and unpack file lab5.zip from Camino. It contains a partially completed program (main5.c), a new pre-compiled library file (libbmp3.a) for manipulating BMP image files, and an associated include file (bmp3.h).

BMP Library: The new library file (libbmp3.a) is identical to the previous library file (libbmp2.a), but eliminates functions GetRGB and PutRGB in favor of providing direct access to the internal data structures that represent an image:

```
typedef struct
           {
           uint8_t    blu ;          // blue component (0 to 255)
           uint8_t    grn ;          // green component (0 to 255)
           uint8_t    red ;          // red component (0 to 255
           } PIXEL ;

typedef struct
           {
           unsigned   rows ;         // image height in pixels
           unsigned   cols ;         // image width in pixels
           unsigned   dpi ;          // dots per inch
           PIXEL      *pxlrow[0] ;   // pointers to pixel rows
           } IMAGE ;
```



The library also includes a new function that will be needed to complete this assignment:

```
IMAGE *NewImage(unsigned rows, unsigned cols) ;
```

Returns a pointer to memory allocated to hold the necessary data structures for an image of size *rows* by *cols*. The actual RGB values of the pixels, however, are not initialized and must be replaced by the user.

Assignment: Complete the source code for each the following two functions that are located within the provided main program (main5.c):

```
IMAGE *HalfSize(IMAGE *image) ;
```

Returns a pointer to a half-size copy of the original image. Each pixel in the new image is computed as the average of a 2x2 array of pixels taken from the original image.

```
IMAGE *DoubleSize(IMAGE *image) ;
```

Returns a pointer to a double-size copy of the original image. Each pixel in the original image is copied four times to fill a 2x2 subarray of pixels in the new image.

Compilation: Compile and link your program using the following command line:

```
gcc –o lab5 main5.c -L. -lbmp3
```

Execution: Execute your program using the following command syntax:

```
./lab5 src-file dst-file {option#}
```

When Done: Demonstrate proper operation of your program to the teaching assistant and upload the completed source code for file main5.c to the lab drop box on Camino. Do not upload any other files.