

# EECS 22: Advanced C Programming

## Week 4

Saeed Karimi Bidhendi  
skarimib@uci.edu

10/18/2017

(c) 2013 Che-Wei Chang

# Review: Provided Function

```
○ #define WIDTH 600      /* Image width */
○ #define HEIGHT 400    /* image height */
○ #define SLEN 80       /* maximum length of file names */

○ int main()
○ {
○     /*
○     * Two dimensional arrays to hold the current image data
○     * One array for each color component
○     */
○     unsigned char  R[WIDTH][HEIGHT];
○     unsigned char  G[WIDTH][HEIGHT];
○     unsigned char  B[WIDTH][HEIGHT];
○     /* Please replace the following code with proper menu */
○     /* with function calls for DIP operations */
○     if(LoadImage("HSSOE.ppm", R, G, B) !=0)
○         {return 10;}
○     /* end of replacing*/

○     return 0;
○ }
```

# Review: Provided Function

```
○ #define WIDTH 600      /* Image width */
○ #define HEIGHT 400    /* image height */
○ #define SLEN 80       /* maximum length of file names */

○ int main()
○ {
○   /*
○    * Two dimensional arrays to hold the current image data
○    * One array for each color component
○    */
○   unsigned char  R[WIDTH][HEIGHT];
○   unsigned char  G[WIDTH][HEIGHT];
○   unsigned char  B[WIDTH][HEIGHT];
○   /* Please replace the following code with proper menu */
○   /* with function calls for DIP operations */
○   LoadImage("HSSOE", R, G, B);
○   SaveImage("Test", R, G, B);
○   /* end of replacing*/

○   return 0;
○ }
```

# Review: Provided Function

- Image Input / Output

- `int LoadImage (const char fname[SLEN],  
                  unsigned char R[WIDTH][HEIGHT],  
                  unsigned char G[WIDTH][HEIGHT],  
                  unsigned char B[WIDTH][HEIGHT]) ;`

- `int SaveImage (const char fname[SLEN],  
                  unsigned char R[WIDTH][HEIGHT],  
                  unsigned char G[WIDTH][HEIGHT],  
                  unsigned char B[WIDTH][HEIGHT]) ;`

- Arguments are passed to the function by reference.

- Use `scanf ("%75s", fname)` to input file name

# Review: Provided Function

- Aging function – as the sample of DIP function

- ```
void Aging(unsigned char R[WIDTH][HEIGHT],
           unsigned char G[WIDTH][HEIGHT],
           unsigned char B[WIDTH][HEIGHT])
{
    int x, y;
    for( y = 0; y < HEIGHT; y++ )
        for( x = 0; x < WIDTH; x++ ) {
            B[x][y] = ( R[x][y]+G[x][y]+B[x][y] )/5;
            R[x][y] = (unsigned char) (B[x][y]*1.6);
            G[x][y] = (unsigned char) (B[x][y]*1.6);
        }
}
```

# Shuffle



- Divide the image into 16 equally sized image blocks
- Randomly choose a pair of image blocks and swap their R,G,B values
- Repeat until all image blocks are swapped
  - An image block may be swapped to its first place again.
- Due to randomness, a different image is created on each run.

# rand()

- Sample Code

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    for(int i = 0; i < 5; i++){
        printf("%d\t", rand() % 10);
    }
    return 0;
}
```

- Generates 5 random numbers between 0 and 9.
- Generates the same set of random numbers in each run.
  - This should be avoided with srand() and time()

# rand() , srand()

- Sample Code

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(){
    srand(time(0));
    for(int i = 0; i < 5; i++){
        printf("%d\t", rand() % 10);
    }

    return 0;
}
```

- Generates 5 random numbers between 0 and 9.
- Generates different set of random numbers in each run.



# Vertical Flip



- For all pixels in the up half picture, swap the color with the pixel in the down half

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 0 | 3 | 5 | 4 | 7 |
| 2 | 1 | 4 | 4 | 3 | 6 |
| 3 | 2 | 5 | 3 | 2 | 5 |
| 4 | 3 | 6 | 2 | 1 | 4 |
| 5 | 4 | 7 | 1 | 0 | 3 |

# Vertically Mirror



- For all pixels in the bottom half of the picture, replace the color to the color of pixel in the top half.

1 4 6  
5 2 1  
7 8 9  
8 2 4  
9 3 7

1 4 6  
5 2 1  
7 8 9  
5 2 1  
1 4 6

# Add Border



- `void AddBorder(unsigned char R[WIDTH][HEIGHT],`
- `unsigned char G[WIDTH][HEIGHT],`
- `unsigned char B[WIDTH][HEIGHT],`
- `int r, int g, int b, int`  
`bwidth);`

Define an aspect ratio of 16:9 (horizontal border thicker)

# Add Border



- `int strcmp(const char *str1, const char *str2)`
  - Used to compare two character strings. Returns 0 if they are the same.
- Border colors:
  - Red (255, 0, 0), Green (0, 255, 0), Blue (0, 0, 255)
  - Yellow (255, 255, 0), Cyan (0, 255, 255), Pink (255, 192, 203)
  - Orange (255, 165, 0), White (255, 255, 255), Black( 0, 0, 0)

# Provided Function

- AutoTest
- test all DIP functions and save the processed image.

```
void AutoTest (unsigned char R[WIDTH][HEIGHT],
               unsigned char G[WIDTH][HEIGHT],
               unsigned char B[WIDTH][HEIGHT])
{
    char fname[SLEN] = "HSSOE";
    char sname[SLEN];

    ReadImage(fname, R, G, B);
    Negative(R, G, B);
    strcpy(sname, "negative");
    SaveImage(sname, R, G, B);
    printf("Negative tested!\n\n");

    ...
}
```

# Provided Function

- AutoTest
- Use the following setting for “ColorFilter” and “AddBorder” functions in your “AutoTest” function:
  - `ColorFilter(R, G, B, 190, 100, 150, 60, 0, 0, 255);`
  - `AddBorder (R, G, B, "orange", 32);`



# Compile/Execute/View/Submit Your Work

- For each DIP options and the AutoTest, a corresponding function **has to** be created for it.
- Compile your program
  - `gcc PhotoLab.c -o PhotoLab -Wall -ansi -std=c99`
- View your processed image
  - <http://bondi.eecs.uci.edu/~youruserid>
- Name your files `negative`, `colorfilter`, `edge`, `vflip`, `vmirror`, `border`, `shuffle` and `bw` for the corresponding function.
- Required files : `PhotoLab.c`, `PhotoLab.txt`, and `PhotoLab.script`.
- In the `PhotoLab.script`, the following commands are required.
  - Compilation of the `PhotoLab.c`
  - Run your `PhotoLab`
  - Use option “`Test all functions`” to test all DIP functions.