Lab 08: Pointers and Functions

Objectives:

- Learn to use pointers in C programs
- Learn the difference between passing arguments by reference and by value

Reading Task 1: Addresses of variables and Pointers in C

Reading Task 2: Returning multiple values from a function

In-Lab Task 1:

For the given program find (and print) the addresses assigned to all the variables against their names.

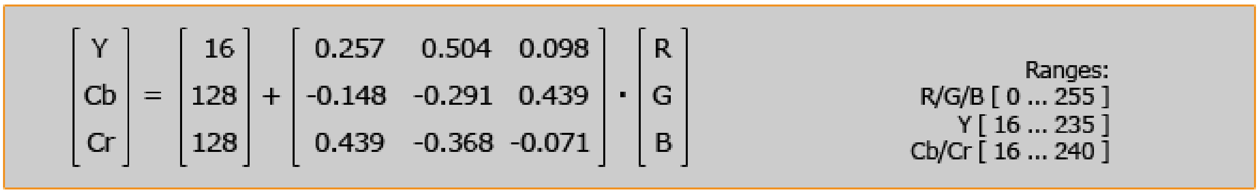
```
#include<stdio.h>
void test(void);

int m = 22, n = 44;
float a = 50.5, b = 8.79;

int main()
{
   int x, y=5, z = -5;
   test();
}

void test()
{
   char ch1 = ‘F’, ch2 = 69, ch3 = 100;
   int x = 5, y = 55, z = 8;
}
```

Code Listing 1

In-Lab Task 2: Color Space Conversion

YCbCr, Y′CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y′ is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

Y′CbCr color spaces are defined by a mathematical coordinate transformation from an associated RGB color space. If the underlying RGB color space is absolute, the Y′CbCr color space is an absolute color space as well; conversely, if the RGB space is ill-defined, so is Y′CbCr.



$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ranges:
R/G/B [0 ... 255]
Y [16 ... 235]
Cb/Cr [16 ... 240]

RGB to YCbCr color conversion for SDTV

Write a C function which takes three integer inputs, corresponding to RGB components of a colored pixel, and coverts them to the YUV color space. The function should have the following prototype.

```
void rgb2ycbcr(int r, int g, int b, float * y, float * Cr, float * Cb);
```

Example:

```
int rgb2gray(int r, int g, int b)
{
      int result;
      result = (int)((0.257*r)+(0.504*g)+(0.098*b));
      return(result);
}
```

Testing your code:

You are given three files:

1. main.c
2. my_lib.h
3. my_lib.o

Complete the function **rgb2ycbcr()** , compile the project using command line and run the code.

```
gcc main.c my_lib.o -o output.exe
```

See if you get the success message.

Post Lab Task: Recursion

Write a recursive version of the Fibonacci function developed in class.