| Week 05, Lab 05 | Weight: 1% | Due: End of your stream's week 5 lab session (via sync) |
|---|---|---|

**Pre-lab Preparation:**

- **Week 1, 2, 3, 4 Lectures, Week 1, 2, 3, 4 Labs**
- **Lecture: Wk05 Day 013, 014**

---

**Lab Activities:**

Remember to **sync** to obtain the lab starting code.

## Exercise 1: Basic array usage

Navigate to the directory: **~/p1.2015s1/lab05/working_copy/ex01/**

Write a program which allocates a ten element **float** array, and assigns to each element in the array the value of the element's index divided by 2**.75f**. Then iterate through the array and print out each element. Open **lab05ex01.c** in the editor and begin! The output must look as follows:

```
+----------------------------------------+
| Element at index [0] is set to 0.000000 |
| Element at index [1] is set to 0.363636 |
| Element at index [2] is set to 0.727273 |
| Element at index [3] is set to 1.090909 |
| Element at index [4] is set to 1.454545 |
| Element at index [5] is set to 1.818182 |
| Element at index [6] is set to 2.181818 |
| Element at index [7] is set to 2.545455 |
| Element at index [8] is set to 2.909091 |
| Element at index [9] is set to 3.272727 |
+----------------------------------------+
```

Be sure to test your program as you develop. Edit, compile and test often as you add features. Remember use **make** to compile your program.

## Exercise 2: `char` arrays

Navigate to the directory: **`~/p1.2015s1/lab05/working_copy/ex02/`**

Write a program which allocates a 52 element **`char`** array. Initialise the first half of the array with the lowercase alphabetic characters **`'a'`** – **`'z'`** and the second half with the uppercase characters **`'A'`** – **`'Z'`**.

Then iterate through the array and print out each element and its corresponding ASCII value. Open **`lab05ex02.c`** in the editor and begin! The output must look as follows:

```
Char at index [0] is set to 'a' its ASCII value is 97
Char at index [1] is set to 'b' its ASCII value is 98
…
Char at index [25] is set to 'z' its ASCII value is 122
Char at index [26] is set to 'A' its ASCII value is 65
Char at index [27] is set to 'B' its ASCII value is 66
…
Char at index [52] is set to 'Z' its ASCII value is 90
```

Be sure to test your program as you develop. Edit, compile and test often as you add features. Remember use **`make`** to compile your program.

## Exercise 3: `char` arrays and using Strings

Navigate to the directory: **`~/p1.2015s1/lab05/working_copy/ex03/`**

Open **`lab05ex03.c`** and add the following code to the file. Replace the comments with appropriate code to achieve the desired action of the comment as described below.

```
#include <stdio.h>

int string_length(char the_word[])
{
    int count = 0;
    while(the_word[count] != '\0')
    {
        count++;
    }
    return count;
}

int count_lowercase_chars(char the_word[])
{
    // Using a loop count how many characters in the_word are
    // lowercase. Return the result.
    return (0);
}

int main(int argc, char* argv[])
{
    char words[][6] = { "Hello", "WoRlD", "And", "Hi!!!", "P1",
                        "Lab05", "*Fun*", "what?", "HELP", "?!?!!" };

    int length_results[10];
    int lowercase_results[10];

    // Generate the results...
    for (int i = 0; i < 10; ++i)
    {
         length_results[i] = string_length(words[i]);
        // Add code to fill lowercase_results with the
        // number of lowercase characters in each string
    }

    // Display the results...
    for (int i = 0; i < 10; ++i)
    {
        printf("\"%s\" is ", words[i]);
        printf("%d characters long ", length_results[i]);
        printf("and contains ");
        // Add code to output the number of lowercase characters
    }

    return (0);
}
```

The output should look as follows:

```
 "Hello" is 5 characters long and contains 4 lower-case characters.
 "WoRlD" is 5 characters long and contains 2 lower-case characters.
```

…

**Once complete, have a TA review your completed exercises 1, 2 and 3 for this lab session. See the end of this document for the review questions.**

## Exercise 4: More character classification

Copy your answer for Exercise 3 into the **`ex04`** directory and rename it **`lab05ex04.c`**.

Modify your code so that it instead produces the following output:

```
"Hello" contains 4 lower-case, 1 upper-case, and 0 non-alphabet glyphs.
"WoRlD" contains 2 lower-case, 3 upper-case, and 0 non-alphabet glyphs.
"And" contains 2 lower-case, 1 upper-case, and 0 non-alphabet glyphs.
"Hi!!!" contains 1 lower-case, 1 upper-case, and 3 non-alphabet glyphs.
"P1" contains 0 lower-case, 1 upper-case, and 1 non-alphabet glyphs.
"Lab05" contains 2 lower-case, 1 upper-case, and 2 non-alphabet glyphs.
"*Fun*" contains 2 lower-case, 1 upper-case, and 2 non-alphabet glyphs.
"what?" contains 4 lower-case, 0 upper-case, and 1 non-alphabet glyphs.
"HELP" contains 0 lower-case, 4 upper-case, and 0 non-alphabet glyphs.
"?!?!!" contains 0 lower-case, 0 upper-case, and 5 non-alphabet glyphs.
```

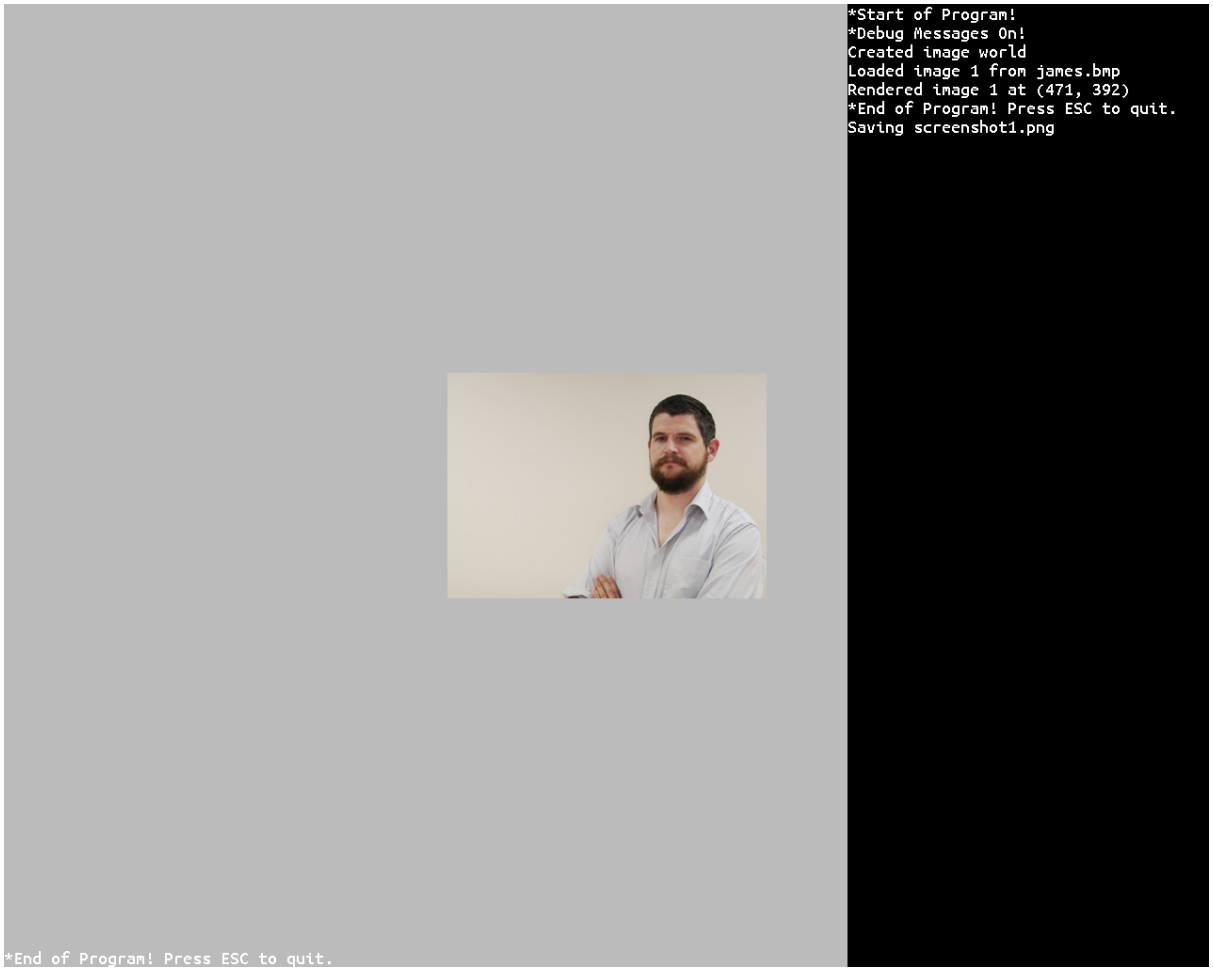
You will need to implement two new functions:

- **`count_upper_case_chars()`**
- **`count_non_alphabet_chars()`**

## Exercise 5: Hey there good looking

Navigate to the directory: **`~/p1.2015s1/lab05/working_copy/ex05/`**

Open **`lab05ex05.c`** and write code to render the following handsome image:



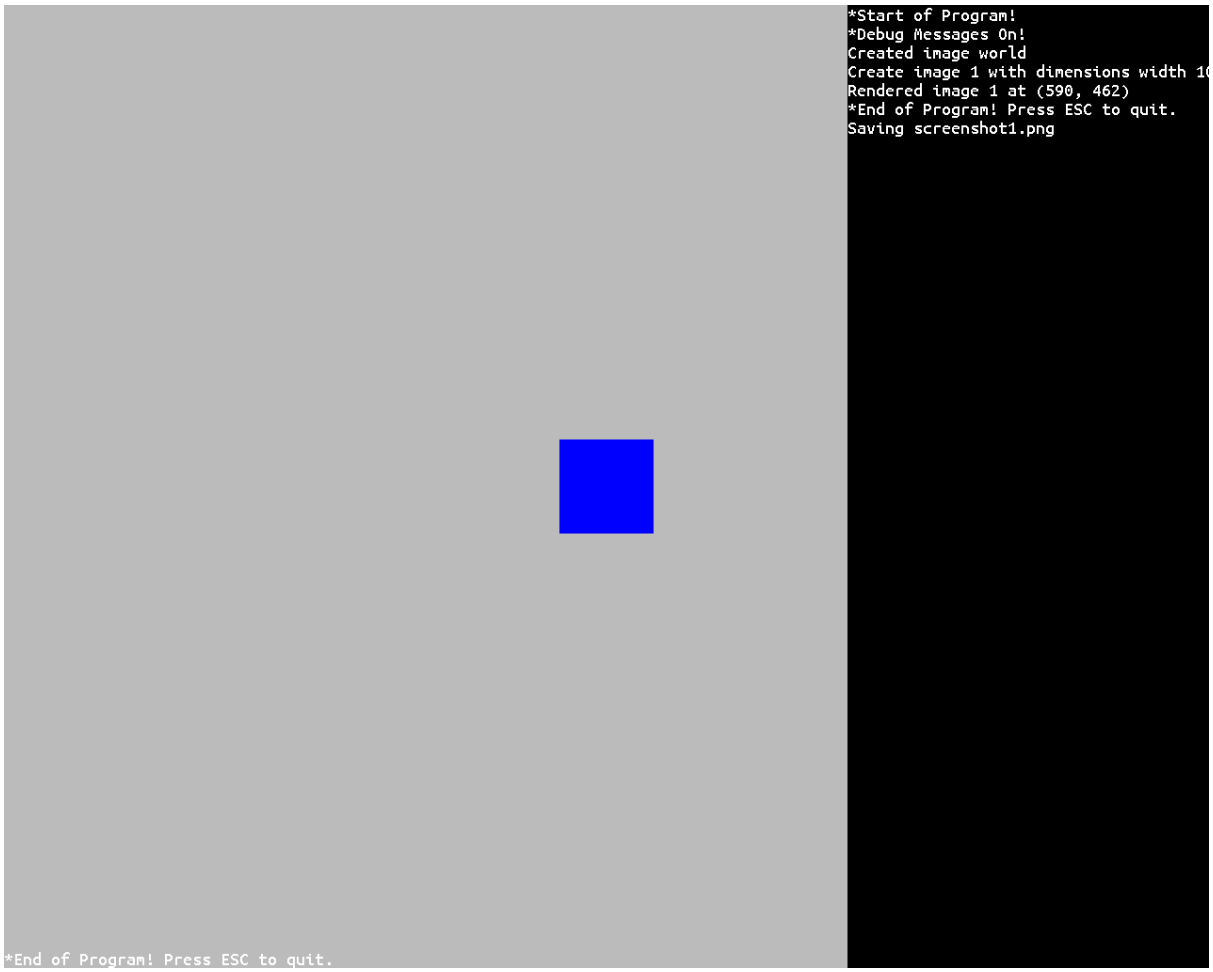

Hints:

- You need to know what the p1 image world functions are for loading a bitmap and putting it to the screen. Prototypes for them are provided in the file **`p1image.h`** in the same directory as the exercise. Try viewing that file with **`less`**.
- Remember that you deal with images by saving an integer handle in a variable. The function for loading an image returns one, and the function for putting an image takes one as an argument.
- To create exactly the image above, you need to put the image in the centre of the screen. But the put_image() function takes the top left coordinate of the image. Where should this be? Hint, the coordinate for the centre of the screen is half the width of the screen and half its height. What is the coordinate for the top-left of the image, if its centre is (0, 0)?

## Exercise 6: Filled rectangle

Navigate to the directory: **`~/p1.2015s1/lab05/working_copy/ex06/`**

Open **`lab05ex06.c`** and have a look at the starting code you have been given. Your first job is to get a blue square to appear on the screen as below.




Look at the code in **`blue_fill()`**. Note that the loop counts over the bytes in jumps of 4, because each pixel is four bytes. Each byte in the pixel represents a different colour component. Therefore:

- **`bytes[i + 0]`** is the blue component
- **`bytes[i + 1]`** is the green component
- **`bytes[i + 2]`** is the red component
- **`bytes[i + 3]`** is the alpha (transparency) component

Once you can fill a blue rectangle, implement the more general **`fill()`** function to fill a rectangle with any colour. Try drawing rectangles of different sizes on the screen in different locations, but be careful not to overlap the edges.

## Week 05, Lab 05 Submission:

Run the **sync** command to submit your completed lab work.

Shutdown your Raspberry PI by pressing **ALT-CTRL-DEL**. Power-down and pack up your Raspberry Pi kit.

## Marking Criteria:

Have you completed each of the following? Have you submitted your code from lab?

| Marking Criteria: | Week 05 Lab 05 Weight 1% | Yes | No |
|---|---|---|---|
| Ex 1: | Ten element float array allocated and used correctly? | | |
| | Output matches requirements? | | |
| Ex 2: | Fifty-two element char array allocated and filled correctly? | | |
| | Output matches requirements? | | |
| Ex 3: | count_lower_case_chars() implemented correctly? | | |
| | Output matches requirements? | | |
| Ex 4: | count_upper_case_chars() implemented correctly? | | |
| | count_non_alphabet_chars() implemented correctly? | | |
| | Output matches requirements? | | |
| Ex 5: | James loaded and displayed? | | |
| | Displayed exactly on screen | | |
| | | | |
| Ex 6: | Able to display a blue rectangle? | | |
| | Able to display any rectangle? | | |
| | Experimentation with size, colour, transparency? | | |

## Next activity: Final Week 5 Lecture and Week 4 and Week 5 Homeworks.