

Exercise: Convert a Grayscale Image into Black and White

In this lesson, you will solve an interesting exercise related to 2D arrays.

We'll cover the following ^

- Problem statement
 - Sample input image
 - Sample output image
- Coding exercise

Problem statement

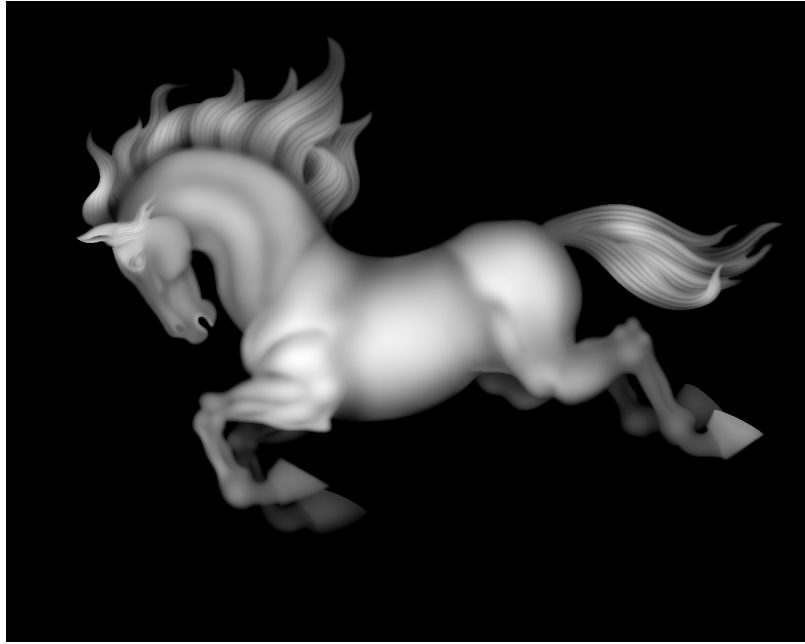
In this exercise, you are given a grayscale image whose pixel values are stored in a 2-D array.

- The `width` specifies the number of columns present in a 2D array.
- The `height` specifies the number of rows present in a 2D array.
- The total number of pixels will be equal to `width*height`.

In the grayscale image, we use a single 8-bit integer to represent the brightness of the pixel. **0** represents black, while **255** represents white. Everything **between 0 and 255** represents different shades of gray.

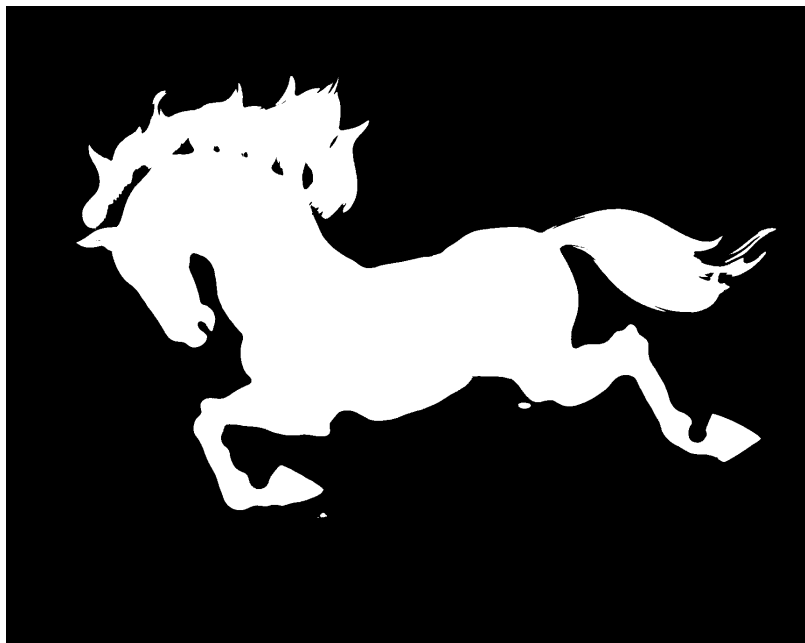
Your task is to apply thresholding to convert a grayscale image to a black and white one.

Sample input image



Sample input

Sample output image #



Sample output


Coding exercise

To solve this exercise, you must know the basics of 2D arrays, loops, and conditional statements.

Before diving directly into the solution, first, try to solve it yourself. We won't check if your code is correct or not. If you get stuck, you can always see the given solution

Solution:

Good Luck! 👍

 **Note:** You don't need to worry about the details of `imagemlib.h`. Just use `height` and `width` parameters to traverse the 2D-array.

```
#include "imagemlib.h"


int main() {
    // Displays input image
    loadFile("input.png");

    // Write your code here

    // Displays modified image
    saveFile("output/modified.png");
}
```



Convert a grayscale image into black and white

 **Note:** In the above code widget, you can see the modified image by pressing the arrow button `>` towards the right of the console

Let's go over the solution review of this exercise in the upcoming lesson.