1. RGB Value for White – 255, 255, 255

RGB Value for Silver - 192, 192, 192

RPG Value for Coral - 255, 127, 80

1. Change in color by lowering red from 255 to 252 is barely noticeable. However, 63 red is almost black, which shows a large difference in color.
2. Since dividing by 2 shifts the removes the leftmost bit, diving by 2 twice will remove the 2 leftmost bits. Therefore, dividing by 4, then multiply by 4 will give the number with the leftmost 2 bits cleared.
3. To remove the 2 rightmost bits, I first used the >> operator to shift the bits to the right twice. This eliminates the 2 rightmost bits, and the number became 45. Then, I used the operator << to shift the bits back in place, and the number became 180.
4. The table is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Original Decimal | Original Binary | Altered decimal after dividing by 4 | Altered binary after dividing by 4 | Altered decimal after multiplying by 4 | Altered binary after multiplying by 4 |
| 183 | 1011 0111 | 45 | 0010 1101 | 180 | 1011 0100 |
| 5 | 0000 0101 | 1 | 0000 0001 | 4 | 0000 0100 |
| 80 | 0101 0000 | 20 | 0001 0100 | 80 | 0101 0000 |

9. I wasn’t able to see any difference from the two pictures. Since clearing the leftmost 2 bits will only affect the RGB value by at most 3, the difference will be very small and barely noticeable.

13. Since the rightmost value is moved to the leftmost, I’ll mod the number by 4 to remove the excessive bits, then shifting the remaining bits to the right 6 times so it’ll be moved to the intended spot

15. To isolate the tens digit, I will divide the number by 10, and then mod 10

To isolate the hundreds, I will divide the number by 100, and then mod 10

To isolate the thousands, I will divide the number by 1000, and then mod 10\

1. Arch doesn’t need to be resized to fit into beach since both its row and column dimension are smaller or equal to the beach’s dimension.
2. Beach will need to be resized before being fit into the other photo. To do so, the columns will need to be reduced from 640 to 360 pixels, which can be done by merging every 2 nearby pixels together.
3. R = 233, G = 172, B = 95
4. R = 64, G = 0, B = 192
5. In order to successfully hide the picture in the source photo, the photo shouldn’t be changed that much and look basically the same. Therefore, putting it at the leftmost 2 bits will be the most effective since the color will barely change while still the necessary data.
6. The rightmost 2 bits is where the majority if the color info is stored. Therefore, it is able to retain the maximum amount of color data while using the least number of bits.
7. After the process of hiding and revealing, only the 2 rightmost bits will be retained for the hidden picture. Therefore, all pixels with similar colors will look the same after the operation, making the picture look very pixelated.

12. Using the rightmost 4 bits instead of just 2 bits will mean that more information can be retained, and therefore the altered image will look closer to the original secret image. However, this will mean that the combined image will look less like the source image, since more color is being altered.

1. int row = Math.random() \* (source.length – secret.length+1);

Int col = Math.random() \* (source[0].length – secret[0].length+1);

1. Yes, it is guaranteed to fit secret within source as source is larger than secret.