APPENDIX Task 3: Template

Picture 10

ROUND 1

First pixel: row 1, column 1, RGB values: (223, 239, 240). We are going to extract the hidden colour values using conversion to binary.

RED: 223 is 1101 1111 in binary

223|2 = 1
111|2 = 1
55|2 = 1
27|2 = 1
13|2 = 1
6|2 = 0
3|2 = 1
1|2 = 1

The four least significant digits are 1111.

We use these are the leading digits of the hidden colour value: 1111 0000

We then convert 1111 0000 to decimal, which gives us 240.

$(11110000)_2 = (1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = (240)_{10}$

And so, the hidden value for red is 240.

GREEN: 239 is 1110 1111 in binary

239|2 = 1
119|2 = 1
59|2 = 1
29|2= 1
14|2 = 0

7|2 = 1
3|2 = 1
1|2 = 1

The least significant digits are 1111.

The hidden colour value is 1111 0000

We then convert 1111 0000 to decimal, which gives us 240.

The hidden value for GREEN is 240

BLUE: 240 is 1111 0000 in binary

The least significant digits are 0000

The hidden colour value is 0000 0000

We then convert 0000 0000 to decimal, which gives us 0.

The hidden value for BLUE is 0.

The hidden colour value is RGB 240, 240, 0.

ROUND 2

Second pixel: row 1, column 2, RGB values: (159, 207, 224). We are going to extract the hidden colour values using conversion to hexadecimal.

RED: 159 is in 9F hexadecimal

159|16 = 15
9|16 = 9

The least significant digit is F.

We use these are the leading digits of the hidden colour value: F0

We then convert F0 to decimal, which gives us 240.

$(F0)_{16} = (15 \times 16^1) + (0 \times 16^0) = (240)_{10}$

And so, the hidden value for red is 240.

GREEN: 207 is in CF hexadecimal

207|16 = 15
12|16 = 12

The least significant digit is F.

We use these are the leading digits of the hidden colour value: F0

We then convert F0 to decimal, which gives us 240.

And so, the hidden value for green is 240.

BLUE: 224 is in E0 hexadecimal

224|16 = 0
14|16 = 14

The least significant digit is 0.

We use these are the leading digits of the hidden colour value: 00

We then convert 00 to decimal, which gives us 0.

And so, the hidden value for blue is 0.

The hidden colour value is RGB 240, 240, 0.

ROUND 3

We can easily decode the images with modulus mathematics, skipping the process converting them to hexadecimals.

Third pixel: row 1, column 3, RGB values: (79, 127, 192).

Firstly, we do modulus 16 operator on the RGB values and we get:

(70|16, 127|16, 192|16) = (15, 15, 0)

Then, we multiply them by 16 and we get:

(15 × 16, 15 × 16, 0) = (240, 240, 0)

Forth pixel: row 1, column 4, RGB values: (15, 47, 80).

Firstly, we do modulus 16 operator on the RGB values and we get:

(15|16, 47|16, 192|16) = (15, 15, 0)

Then, we multiply them by 16 and we get:

(15 × 16, 15 × 16, 0) = (240, 240, 0)

The reason this works is because we convert decimal numbers to hexadecimal using modulus 16 operator until we are left with zero. Using modulus operator once in essence give us the last digit of the hexadecimal value. We would then multiply this number with 16 to simulate adding a zero to the last hexadecimal digit

RESULTS FROM MY IMAGE

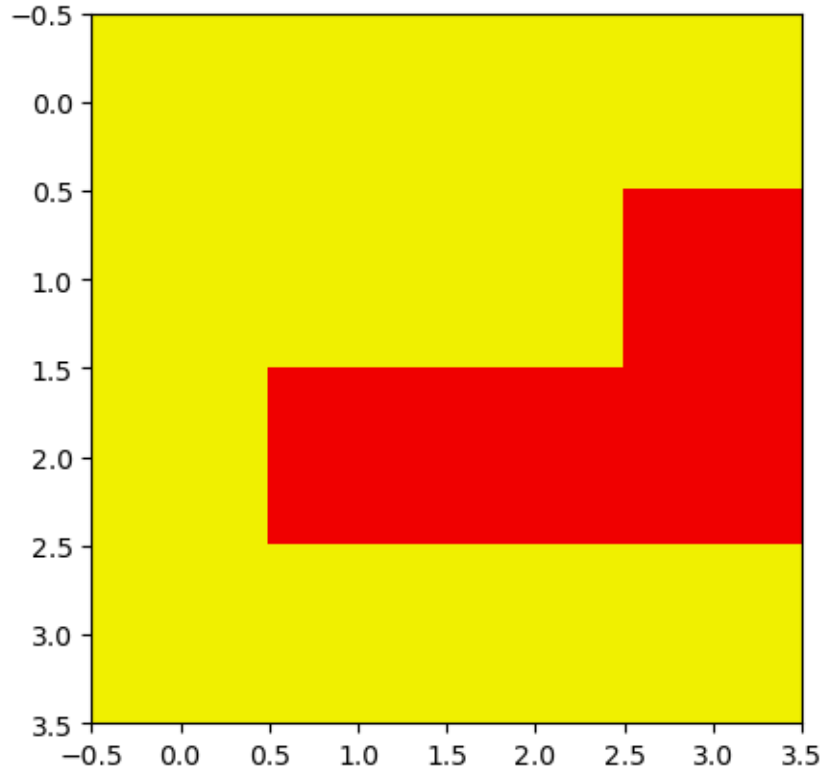
My initial image and colour values:

Image # 10

223 239 240	159 207 224	79 127 192	15 47 80
159 207 224	79 127 192	15 47 80	15 32 80
159 207 224	79 112 192	79 112 192	15 32 80
223 239 240	79 127 192	223 239 240	191 223 224

Hidden image colours values and image:

```
R  [[240 240 240 240]
 [240 240 240 240]
 [240 240 240 240]
 [240 240 240 240]]
G  [[240 240 240 240]
 [240 240 240   0]
 [240   0   0   0]
 [240 240 240 240]]
B  [[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
```



Corresponding Letter: E

RESULTS FROM ALL IMAGES

Letters in order: GRACEHOPPER

I wrote python code to solve all the images. Took me the whole day. Hexadecimal turns out to be a pain to work with. You can find the code

on jupyter notebook here. I didn’t think of skipping converting to hexadecimal and just work with modulus 16 until I wrote this.