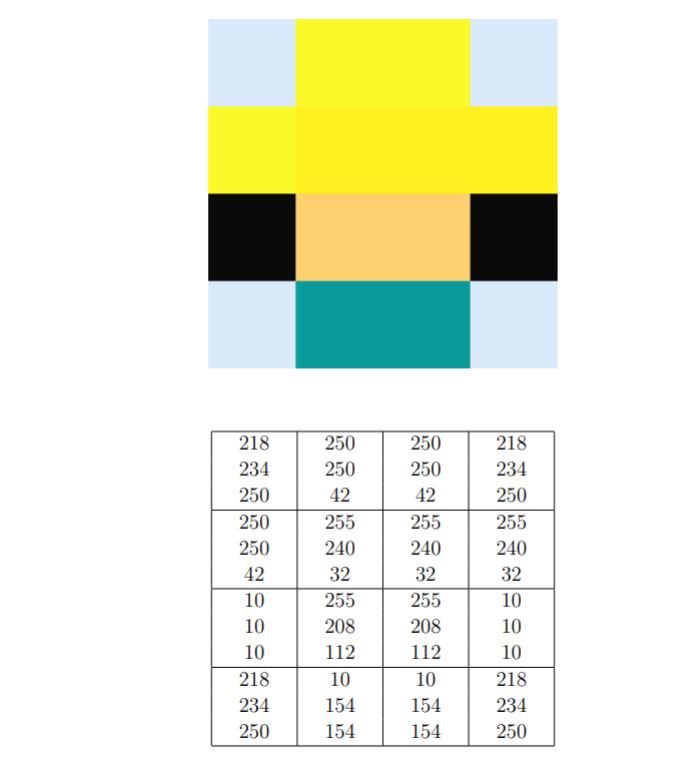
**Steganography**

Hey reader! Hello 😊

My name is Naazleen.

I will take First Image as example and explain

Image#1:



**ROUND – 1:**

**Using Binary conversion:**

**Pixel 1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****Colours**** | ****Decimal**** | ****Binary**** | ****Decoded Binary**** | ****Decoded Decimal**** |
| ****R**** | **218** | **1101 1010** | **1010 0000** | **160** |
| ****G**** | **234** | **1110 1010** | **1010 0000** | **160** |
| ****B**** | **250** | **1111 1010** | **1010 0000** | **160** |

 **---------------------------------------------------🡪** 

**(218, 238, 250) (160, 160, 160)**

**Explanation:**

* **First, we should convert the R, G, B values to binary**
* **By performing division by 2, and storing the remainder and then reverse the remainders (bottom to top)**

|  |  |
| --- | --- |
| ****Division**** | ****Remainder**** |
| ****218 ÷ 2 = 109**** | **0 ^** |
| ****109 ÷ 2 = 54**** | **1 |** |
| ****54 ÷ 2 = 27**** | **0 |** |
| ****27 ÷ 2 = 13**** | **1 |** |
| ****13 ÷ 2 = 6**** | **1 |** |
| ****6 ÷ 2 = 3**** | **0 |** |
| ****3 ÷ 2 = 1**** | **1 |** |
| ****1 ÷ 2 = 0**** | **1 |** |

**So, we get 1101 1010**

* **Then we take low precedence values, i. e, 1010 ad concatenate 0000**
* **We get the binary value 1010 0000**
* **Come on, Now let’s convert it again to decimal**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |

**Add all the one’s with place holder values, we get**

**218 + 32 = 160**

* **Similarly, if we do for G, B we get 160 and 160**
* **So, decoded pixel value is R – 160, G – 160, B – 160**

**ROUND – 3:**

**Using Hexadecimal conversion:**

**Pixel 2:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****Colours**** | ****Decimal**** | ****Hexadecimal**** | ****Decoded Hexadecimal**** | ****Decoded****  ****Decimal**** |
| ****R**** | **250** | **FA** | **A0** | **160** |
| ****G**** | **250** | **FA** | **A0** | **160** |
| ****B**** | **42** | **2A** | **A0** | **160** |

 **-----------------------------------------------------🡪** 

**(250, 250,42) (160, 160, 160)**

**Explanation:**

* **First, we should convert the R, G, B values to hexadecimal**
* **By performing division by power of 16 which less than the value, and storing the quotient**
* **Then subtract power of 16 X quotient and subtract from original value and repeat it until we get 0.**

|  |  |  |
| --- | --- | --- |
| **Divide** | **Subtract** | **Quotient** |
| **250** **÷ 16 = 15** | **250 - 16 X 15 = 10** | **15 -> F** |
| **10 ÷ 1 = 10** | **10 - 1 X 10 = 10** | **10 -> A** |

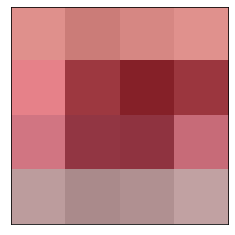
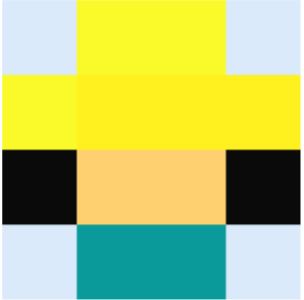
**So, take from top to bottom, we get FA**

* **Take the least significant digit is A, take A and concatenate with 0 we get A0**
* **Value of A0 is**

|  |  |
| --- | --- |
| **16** | **1** |
| **A -> 10** | **0** |

* **10 X 16 + 1 X 0 = 160**
* **Similarly, If we convert G, B we get 160, 160.**

**If we continue this process and repeat for every pixel, we get decoded image**



# **ORIGINAL HIDDEN**

* Woohoo! We decoded the image, the letter is G

**ROUND – 3:**

* As we already know the hidden RGB values of the colour codes in ROUND 1 and 2, we don’t need to do further binary conversions for the same RGB values Binary values are obtained by finding the remainders of the number by division of 2.
* If we continue and repeat this process to each and every image then we get



The answer to this would be



**GRACEHOPPER**! Shh … SECRET