|  |  |
| --- | --- |
| ***Programming Language*** | *Python Matlab Octave* |
| ***Programming Environment*** | *Visual Studio Code(Pillow) /MATLAB R2021a* |
| ***Secret Key*** | *Message: $If you want to find the secrets of the universe, think in terms of energy, frequency and vibration.* |
| ***Reflections*** | *In the code of the message, I had difficulty converting each 8-bit chunk of the binary message to its corresponding ASCII character, but I learned how. In Matlab I learned to reverse the message thread and*  *display it using the 'disp' function.* |

***Source Code***

# Matlab Code:

sentinel='$';

I = imread('21.tif'); [height, width, ~] = size(I); pixels = reshape(I, [],1); pixels = double(pixels);

lsb\_plane = bitget(pixels(:,1),1); bits = reshape(lsb\_plane', 1, []);

message = '';

for i = 1:8:length(bits) byte = bits(i:i+7); byte\_str = num2str(byte);

byte\_ascii = bin2dec(byte\_str); char = native2unicode(byte\_ascii); message = [message char];

if char == sentinel break;

end

end

reversedMessage = reverse(message); disp(["Message: " + reversedMessage]);

# Python Code:

from PIL import Image

# Define the input parameters

img\_path = "C:/Users/Emine/Desktop/Python\_code/21.tif" sentinel = "$"

# Open the image and convert it to a 2D array of pixels img = Image.open(img\_path)

pixels = list(img.getdata()) width, height = img.size

pixels = [pixels[i \* width:(i + 1) \* width] for i in range(height)] # Extract the LSB bitplane

lsb\_plane = [[pixel & 1 for pixel in row] for row in pixels]

# Convert the bitplane to a 1D array of bits in column-major order bits = [lsb\_plane[j][i] for i in range(width) for j in range(height)]

# Read the message in column-major order, every 8 bits message = ""

for i in range(0, len(bits), 8):

byte = bits[i:i+8]

byte\_str = "".join(str(bit) for bit in byte) byte\_ascii = int(byte\_str, 2)

char = chr(byte\_ascii) message += char

if char == sentinel:

break

# Print the message reversed\_message = message[::-1] print("message: "+ reversed\_message)