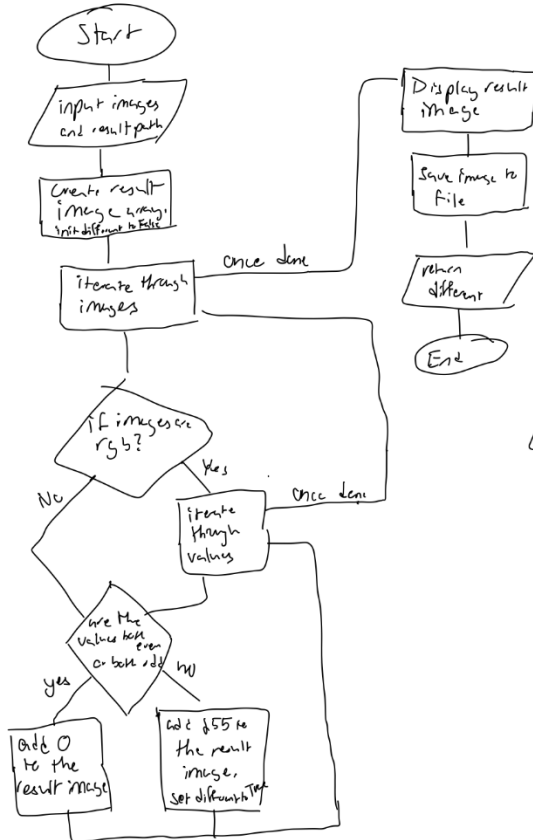# TP 2, task 1

compare_images ()

Start

↓

input images and result path

↓

create result image array, init different to False

↓

iterate through images

— once done →

if images are rgb?

- No
- Yes →

iterate through values

— once done →

are the values both even or both odd?

- yes
- no

Add 0 to the result image

add 255 to the result image, set different to True

Display result image

↓

save image to file

↓

return different

↓

End

main ()

Start

↓

input paths for images and output

↓

load images

↓

are images different shapes?

- Yes
- No

Raise Error

compare images

↓

what did it return

- True
- False

Print that images are different

Print that images are the same

↓

End

```
                    ( START )
                        |
                        v
            / Plain text  input /
                        |
                        v
            / Key  input /
                        |
                        v
            / Start/End Segs /
                        |
                        v
         [ Convert Plain into # ]
                        |
                        v
         [ Convert Key into # ]
                        |
                        v
         [ add
           increment key to plain ]
                        |
                        v
            / Print new part /
                        |
                        v
            / Print Code Binary /
                        |
                        v
                    ( END )
```

```
start
```

Import matplotlib, PIL, and the load image function

Def analysis

Check if image is color; get 3 dimensions if true

Check if image is grayscale: get 2 dimensions if true

Check if dimensions are valid: if false print an error

Find a specific row and colum value in the image, divide it by 2 and take the remainder

Check if remainder is between 0 and 255

Check if the binary value if the same as the remainder; if the remainder is zero iterate up; if the remainder is not zero, iterate down

Plot and save figure

Def main

Input message to encrypt, cypher key, start and end sequences, image path, and encoded message path

Run def analvsis

Run imported encryption function

Print outputs

end