

**Cairo University,**

**Faculty of Computers and Artificial Intelligence**

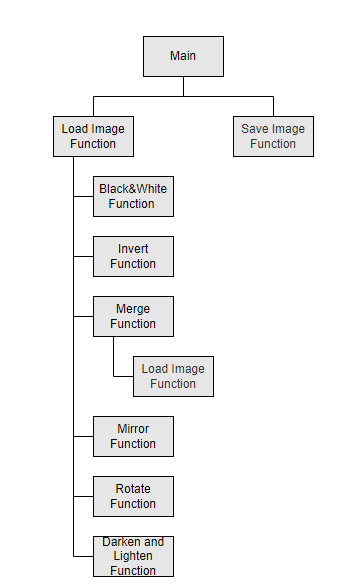
**CS112 – Structured Programming**  
**Second Semester 2021-2022  
s**

**Assignment #3 Report**

**Marwan Ahmed – 20210377**

**Mohamed Waleed Mohamed Abd El-Wahab – 20210363**

**Function Decomposition Diagram**



**Invert Filter Algorithm:**

For each row:

For each column:

Pixel [row][column] = 255 – Pixel[row][column]

**Rotate Filter Algorithm:**

Transpose the matrix:

For i = 0 (in range size of pixels):

For j = i (in range size of pixels):

swap (Pixel [i][j] with Pixel [j][i])

Then we flip the matrix horizontally:

For i = 0 (in range size of pixels):

For j = 0 (in range size of pixels):

swap (Pixel[i][j] with Pixel[i][SIZE – j -1]

**Black & White Filter Algorithm:**

Calculate the average brightness of every pixel by using “average += image[i][j]” then dividing the average by the total pixels “average/ 256 \* 256”

Then for every pixel, if its value is > average, we change it to 255, otherwise if its lower we change it to 0.

**Mirror Filter Algorithm:**

To mirror the right or left side of the image we have to make the “image[i][j] = image[i][255 – j]” so that they would have the same value but reversed till they reach the middle.

And for the bottom or upper sides we use “image[i][j] = image[255 – i][j]” to do the same thing.

The only difference is we’re going to mirror the upper or left halves of the image “[i] & [j] must = 255” so that they would start from bottom or right sides of the image, and for mirroring the bottom or right sides we “[i] & [j] must = 0” so that they would start from the upper or left halves of the image.

**Merge Filter Algorithm**

Load another image

For i=0 (in range size of pixels):

For j=0 (in range size of pixels):

Image[i][j] = ( image1 [i][j] + image2 [i][j] ) / 2