# LAB # 04:

**Lab Objective:**

The objective of this lab is to implement some basic image processing arithmetic including addition and multiplication.

**Lab Description:**

## Arithmetic/logic operations involving images are performed on a pixel-by-pixel basis between two or more images. As an example, subtraction of two images results in a new image whose pixel at coordinates (x, y) is the difference between the pixels in that same location in the two images being subtracted. Depending on the hardware and/or software being used, the actual mechanics of implementing arithmetic/logic operations can be done sequentially, one pixel at a time, or in parallel, where all operations are performed simultaneously.

## Some Useful Commands:

1. To calculate sum of two arrays: my\_sum = **array\_1 + array\_2**
2. To change data type of an array: my\_ array = my\_array.astype(np.float64)

## Lab Tasks:

**1:** Create a distance map of 480 rows and 640 columns as shown below, The image has 0 at the corners and 255 at the middle.

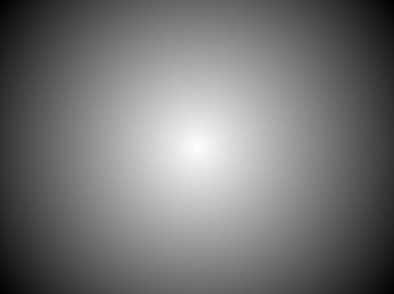


P (i, j) =255 – (r / c \* 255)

Where: c = distance between location (0, 0) and center of image.

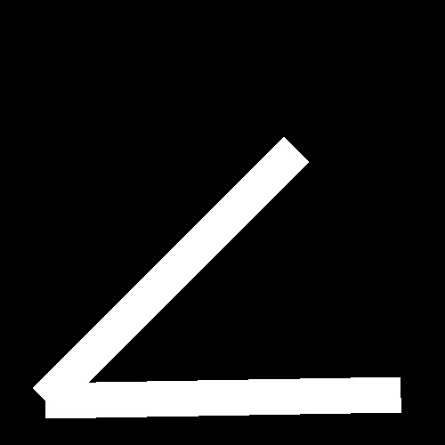
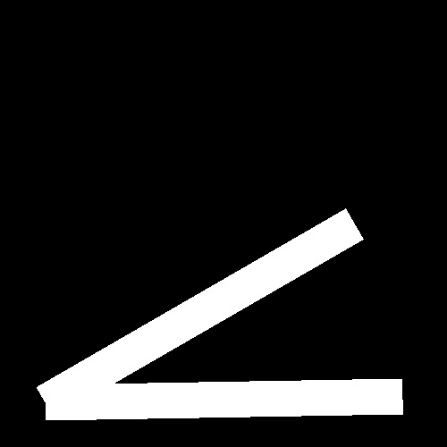
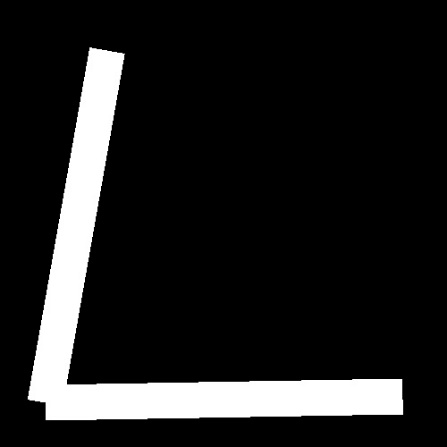
r = distance between pixel location and center of image.

**2:** Multiply the image obtained in Task 1 with image of the balloon. Take necessary measures for data types and ranges.

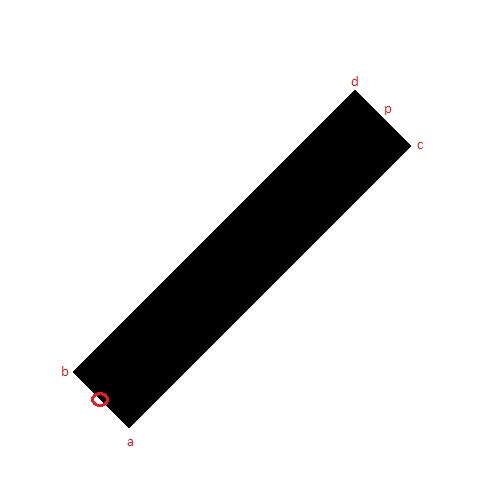
X =

**3:** Write a code that take theta from the user and draw a rectangle in that direction as shown. The size of the image should be 500 x 500, origin of the angle is at location (i, j) = (450, 50) and, length and width of rectangle is equal to 400 and 20 pixels respectively. Description is on the next page.

Theta = 45 theta = 30 theta = 80

**Description:** let point x is at origin O shown with a circle in the image below. Let the rectangle has length l and width w.



Then the equation of corners will be as follows:

ax = x + (w \* cos(t)) ; The t is in radians

ay = y + (w \* sin(t))

bx = x - (w \* cos(t))

by = y - (w \* sin(t))

px = x - (l \* sin(t))

py = y + (l \* cos(t))

cx = px + (w \* cos(t))

cy = py + (w \* sin(t))

dx = px - (w \* cos(t))

dy = py - (w \* sin(t))

Now a point to lie inside the rectangle must meet the following four conditions.

* j < ((i-ax)/tan(-t))+ay:
* j > ((i-bx)/tan(-t))+by:
* j > ((i-ax)\*tan(t))+ay:
* j < ((i-cx)\*tan(t))+cy: