ASSIGNMENT 01

PAPER: IMAGE PROCESSING

NAME: MANASH CHOUDHURI

ROLL NO: CS-13

Q1) Write an pyhton program to convert an RGB image to Binary image.

Ans:

```
# python program for convert an RGB image to a binary image

import cv2

#  read the image file as gray scale image
img = cv2.imread('m01.jpg', 2)

# set the threshold value
ret, bw_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# show the image
cv2.imshow("Binary Image", bw_img)

# waitKey and destroyAllWindows method to keep our window always open until we
#press any key or close our window
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input:

Output:

Q2) Write an python program to convert an RGB image to Gray Scale image.

Ans:

```
# python program for convert rgb image to gray scale image
import cv2

#read the image
img_rgb = cv2.imread('m02.jpg')

# Use the cvtColor() function to grayscale the image
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)

#show the image
cv2.imshow('Gray image', img_gray)

# waitKey and destroyAllWindows method to keep our window always open until we
#press any key or close our window
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input:

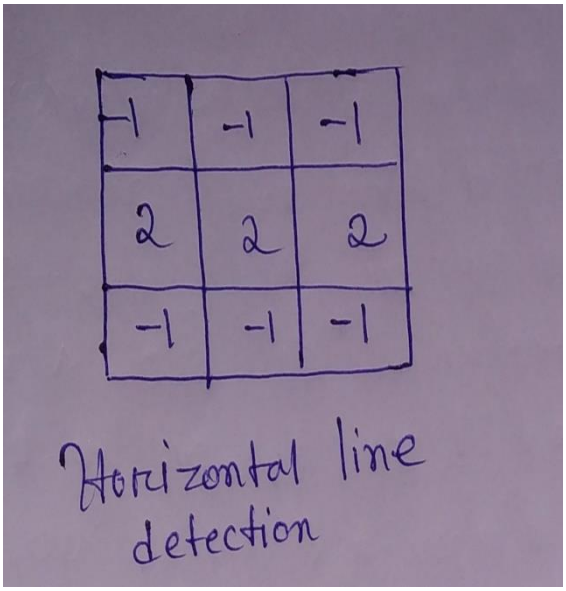
Output:

Q3) Write an python program to detect line of an given image.

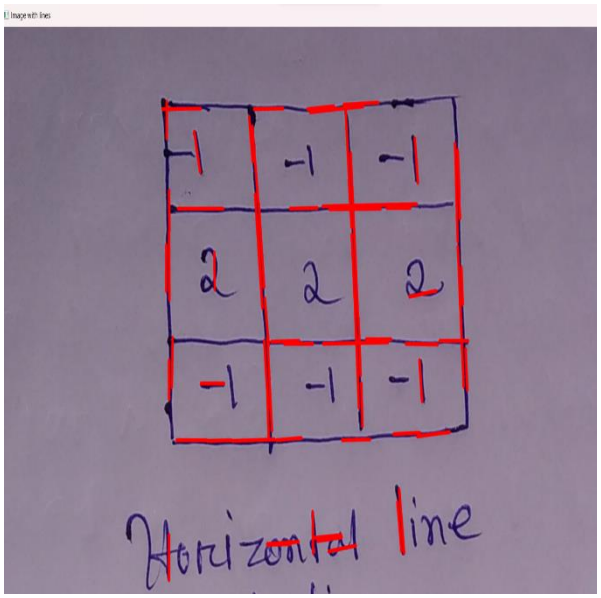
Ans:

```
# pyhton program to detect lines


import cv2
import numpy as np

src_img = cv2.imread('l01.jpg')

dst_img = cv2.Canny(src_img, 50, 200, None, 3)

linesP = cv2.HoughLinesP(dst_img, 1, np.pi / 180, 50, None, 50, 10)

for i in range(0, len(linesP)):
            lin = linesP[i][0]
            cv2.line(src_img, (lin[0], lin[1]), (lin[2], lin[3]), (0,0,255),
3, cv2.LINE_AA)

cv2.imshow("Image with lines", src_img)
cv2.waitKey(0)
```

Input:



Output:



Q4) Write an pyhton program to detect Edge of an given image.

Ans:

```
# pyhton program for edge detection

import cv2

# Read the original image
img = cv2.imread('e01.jpg')

# Convert to graycsale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Blur the image for better edge detection
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

# Canny Edge Detection
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)
# Display Canny Edge Detection Image
cv2.imshow('Edge Detection', edges)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input:

Output: