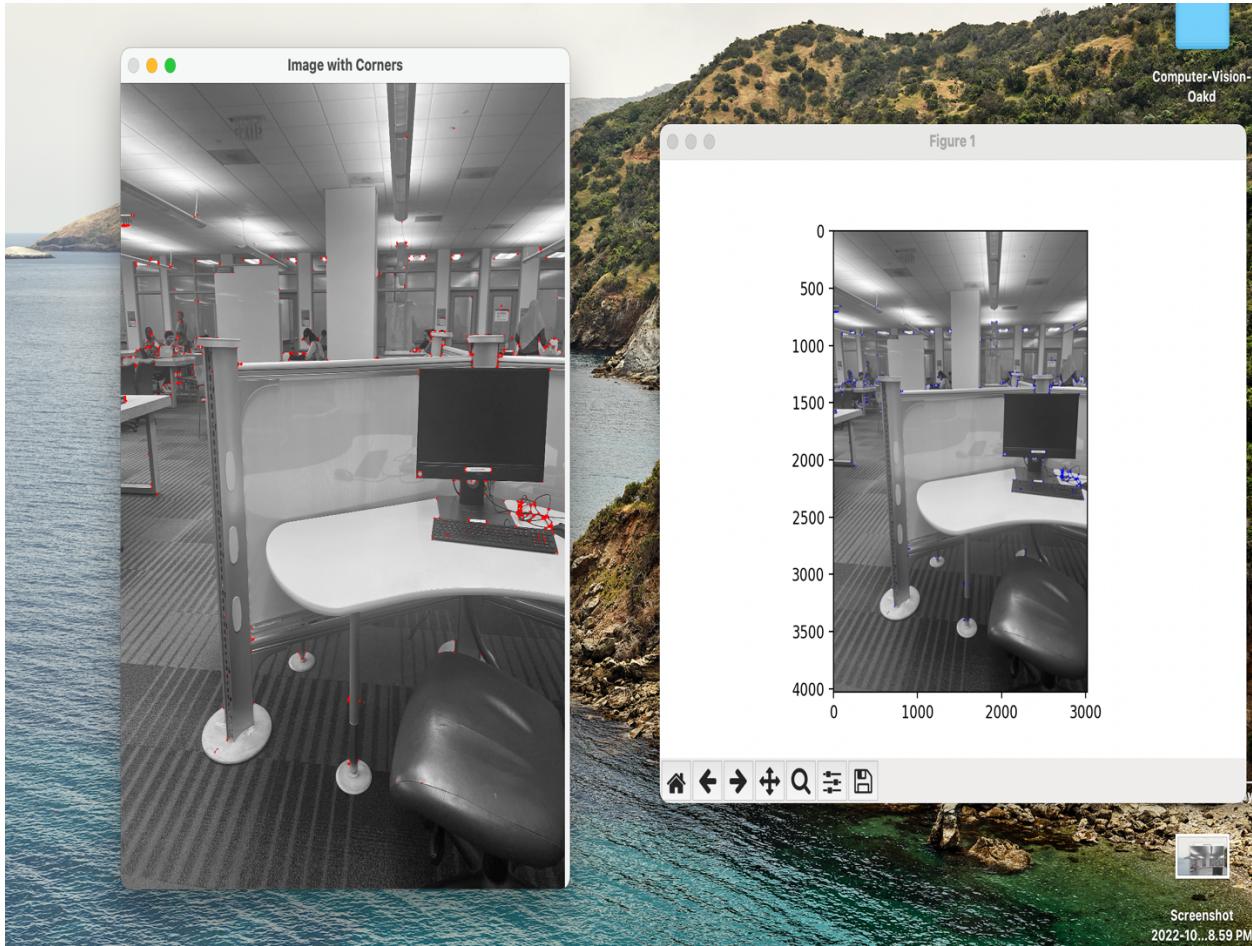
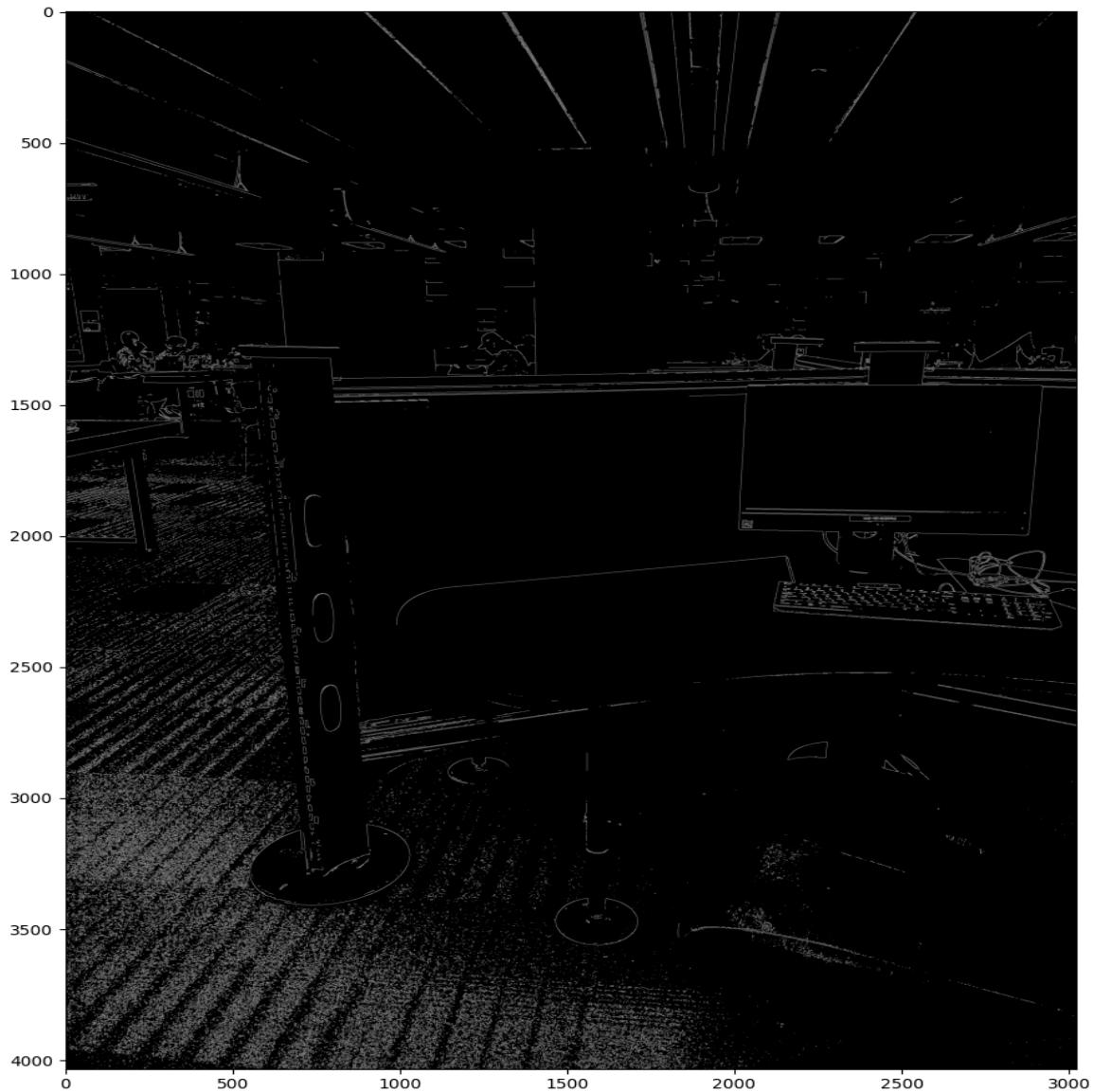


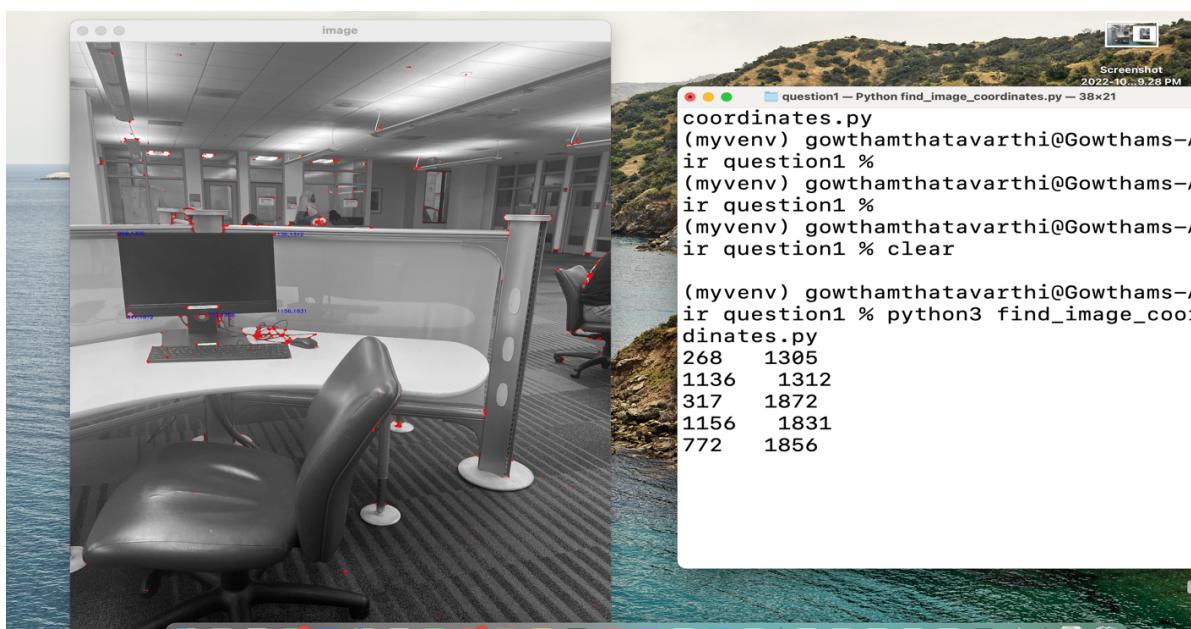
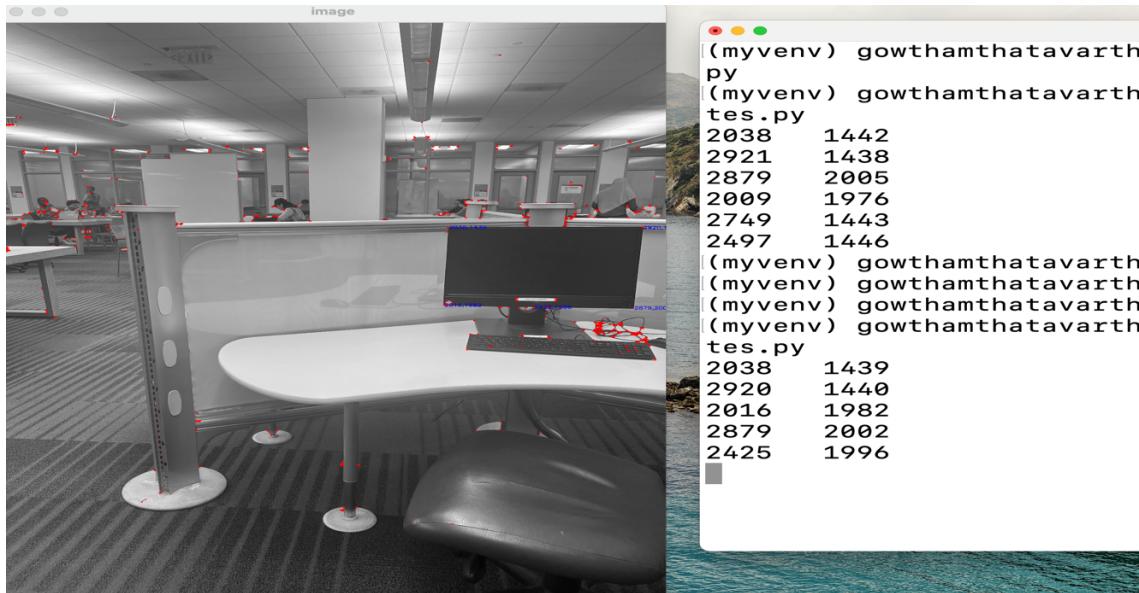
1a) Taken a video and picked two images from the video and placed in images folder. The **harris_detection_a.py** script in question2 folder and it takes img1.png image from images folder and converts it to grayscale and initially it converts the image into float32 and then applies inbuilt **Harris corner detection**. The output is written into img1c.png. Below is the output on execution.



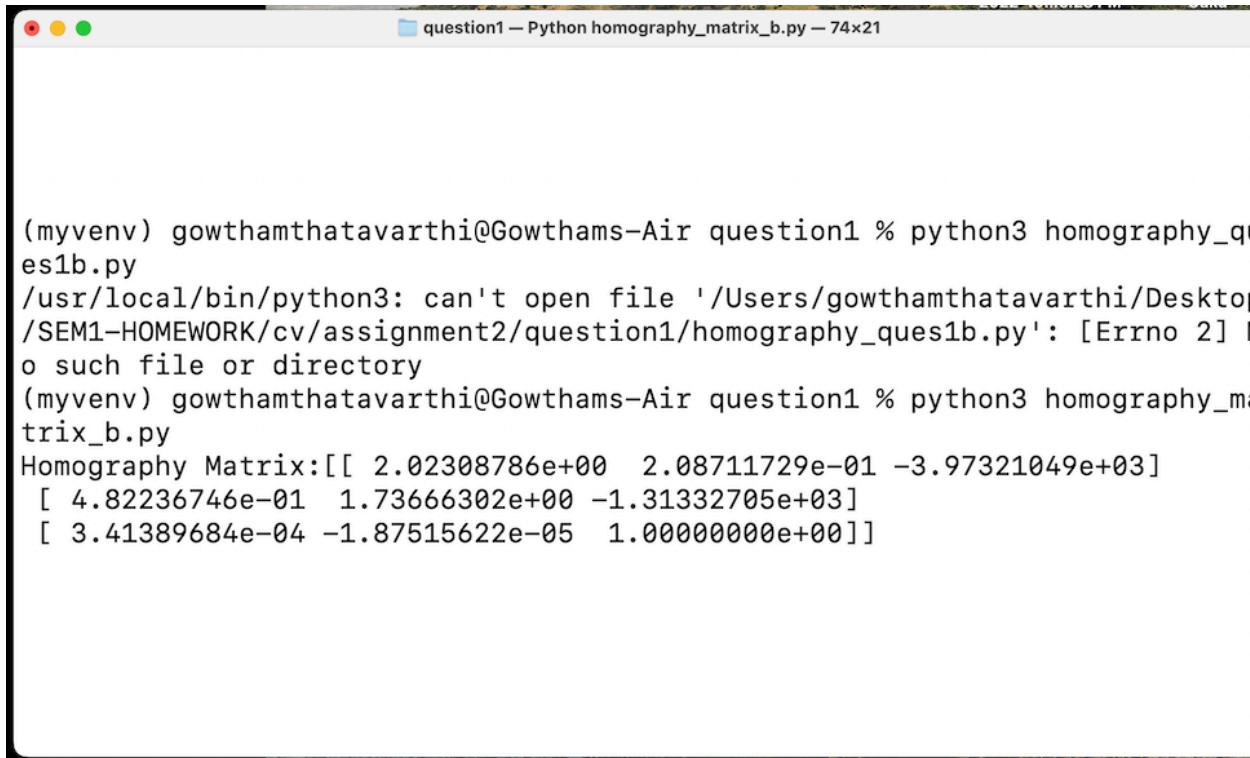
1a) Canny edge detection: The canny_edge_detection.py script and it takes img1.png which is in grayscale image from images folder and initially it converts the image and then applies inbuilt **canny edge detection algorithm**. The output is written into canny_edge_output.png. Below is the output on execution.



1b) Another image img2C.png is taken and image_coordinates_detector.py script is used to get 5 corners of the monitor object used in the image. The following is output after execution



Used this coordinates and run script **homography_matrix_b.py** to get output. One execution the below output is got.



```
(myvenv) gowthamthatavarthi@Gowthams-Air question1 % python3 homography_qu  
es1b.py  
/usr/local/bin/python3: can't open file '/Users/gowthamthatavarthi/Desktop/  
/SEM1-HOMEWORK/cv/assignment2/question1/homography_ques1b.py': [Errno 2] N  
o such file or directory  
(myvenv) gowthamthatavarthi@Gowthams-Air question1 % python3 homography_ma  
trix_b.py  
Homography Matrix:[[ 2.02308786e+00  2.08711729e-01 -3.97321049e+03]  
 [ 4.82236746e-01  1.73666302e+00 -1.31332705e+03]  
 [ 3.41389684e-04 -1.87515622e-05  1.00000000e+00]]
```

The calculated Homography matrix

```
[[ 2.02308786e+00  2.08711729e-01 -3.97321049e+03]  
 [ 4.82236746e-01  1.73666302e+00 -1.31332705e+03]  
 [ 3.41389684e-04 -1.87515622e-05  1.00000000e+00]]
```

2. run the **cv2_sticher.py** script in question2 folder which takes input of all the five 3 set images and runs cv2 image stitcher method and outputs images to stichedSet1.png to stichedSet5.png in question2 folder. The output on execution is below.

Input three images of 1 set are:

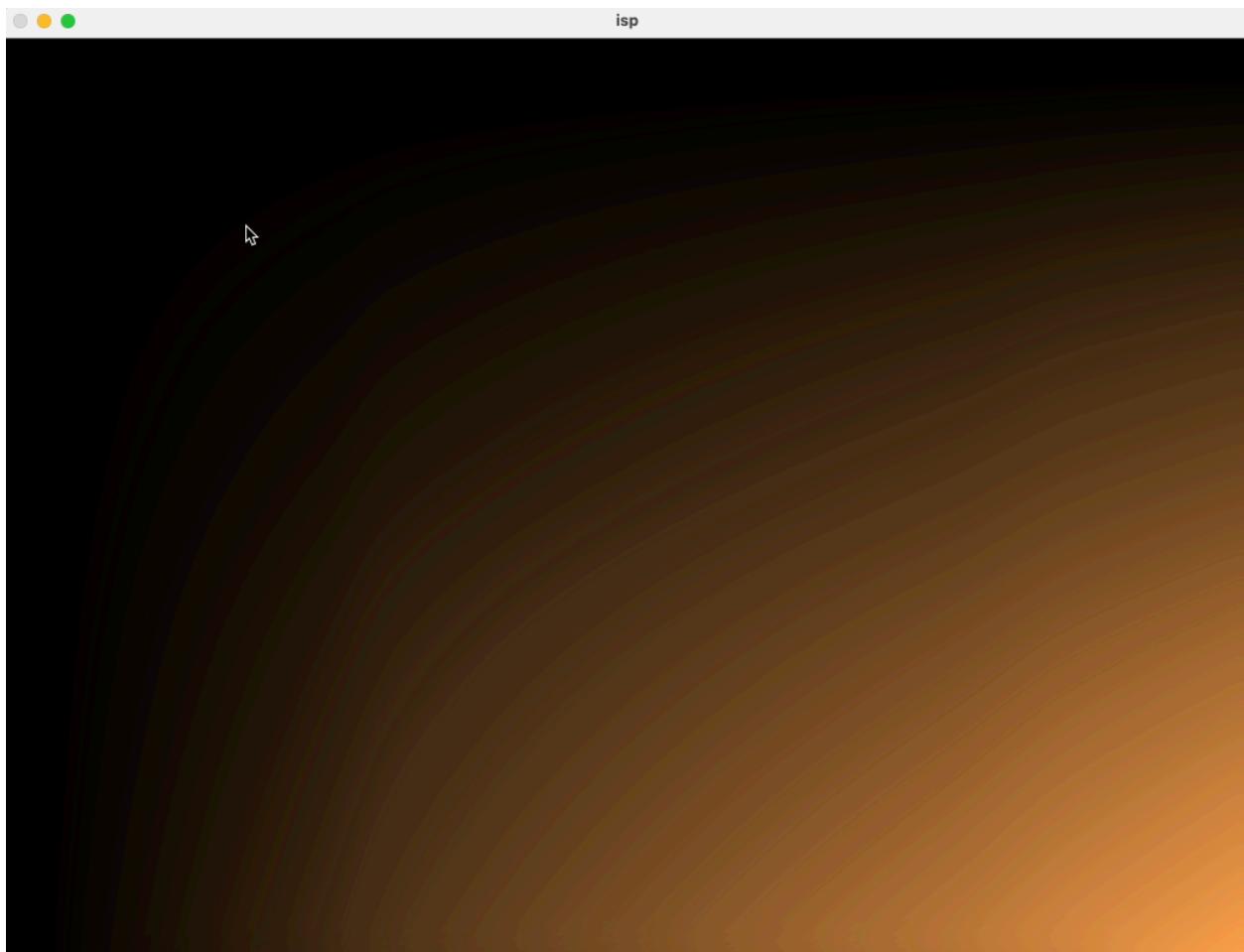




2. The output image on execution

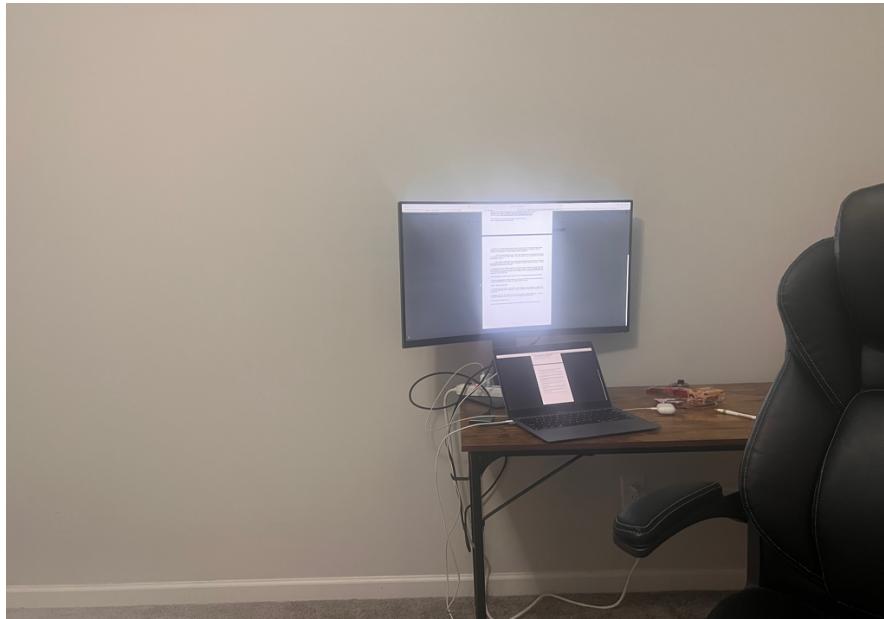


3. In question3 folder, the script **part3.py** file **integral image function** calculates the integral values of image which is custom written. The script has been run the video feed has been screen recorded and attached **question3.mov**. a image snippet of feed is given below.

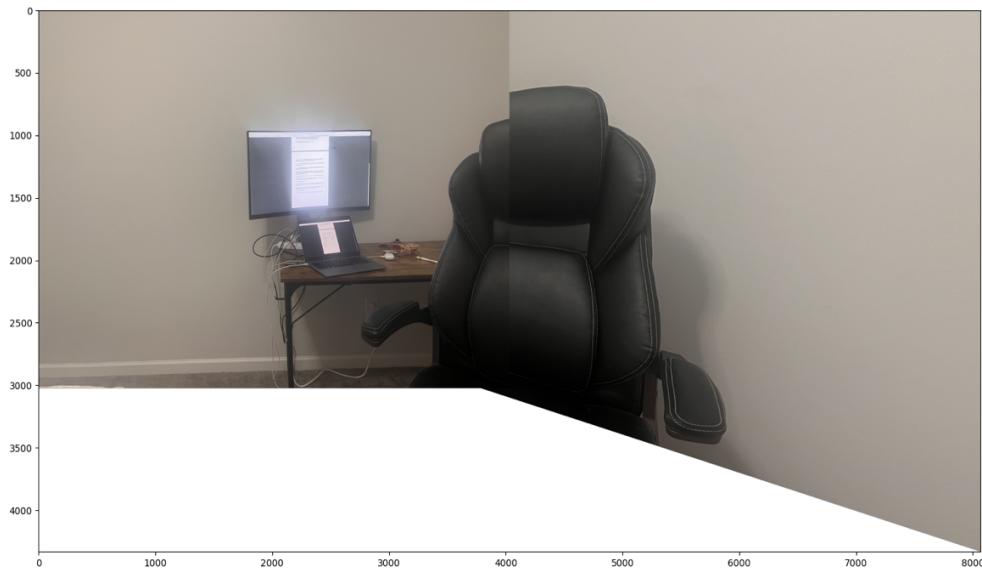


4. In question4 folder, the script **img_stitcher_sift.py** on execution gives the stitched image **sift_ouput.png**. The main logic is to find the the main feature points using cv2 **sift algorithm** in **Main_Ponts_Func()**. Then, **Match_The_Key_Points_Func()** uses KNN algorithm to find nearest possible match between images. Then, use **Find_Homography()** method to find homography and use it to stitch the image using **cv2.WarpPerspective()** function to get the result image. The output on execution is below.

These are the input images present in images folders in question4 and question5 folders.

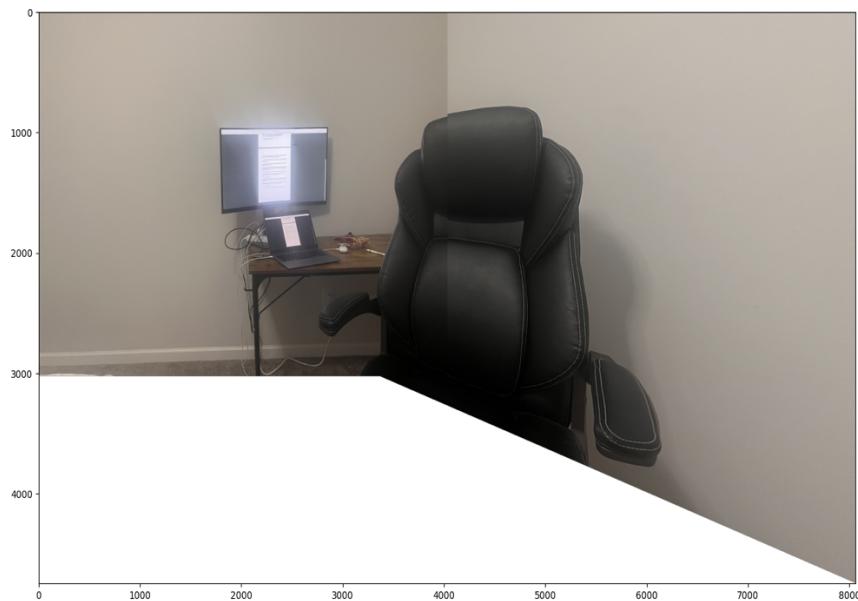


STITCHED IMAGES



SIFT --- 180 degrees panoramic view

5. In question5 folder, **img_stitcher_orb.py** script is same to sift script but the **Main_Points_Func()** uses cv2 orb feature detection



ORB --- 180 degrees panoramic view