Q2.

1.

- First I have converted the image to gray scale and then computed keypoint and descriptors.
- Then I have put the keypoints on both the images using the snippet :

```
keypoint_image1 = cv2.drawKeypoints(image1, keypoints1, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
keypoint_image2 = cv2.drawKeypoints(image2, keypoints2, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

- Then plotted the images:

2.

- I have initialized the brute force feature matcher and flann feature matcher.
- Performed k nearest neighbor matching between two descriptors using both brute force and Flann both.
- Using the ratio test to enhance match quality, a set of descriptor matches is filtered out of trustworthy matches by keeping only those where the best match distance is much smaller (less than 75%) than the second-best match distance. - For both Flann and Brute force.
- Used OpenCV to visualize the good matches between keypoints from two images by drawing lines between them, excluding any unmatched single keypoints, which enhances the clarity and usefulness of the visual output. - For both Flann and Brute force.

3.

- extracted and reshaped the coordinates of the matched keypoints from the first image(image1) and second image (image2) based on the good matches identified previously (good_matches_bf). (point1 and point2)
- Then calculated the left and right homography matrix between point1 and point2.

```
[[-5.38798676e+01  9.96645125e-01  1.94995168e+04]
 [-1.63439985e+01 -3.34163576e+01  1.01803863e+04]
 [-5.59262109e-02 -1.86804273e-03  1.00000000e+00]]
```

```
And
```

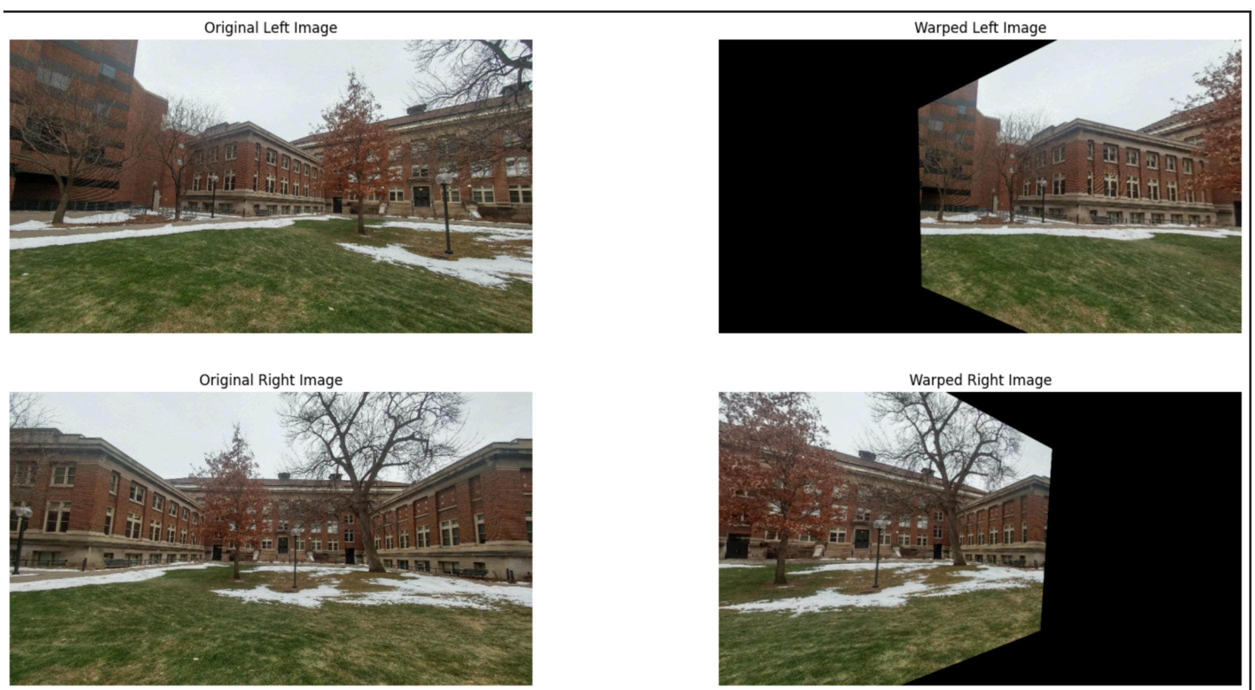
```
[[-1.79925616e-02 -1.82101299e-02  3.65665156e+02]
 [-3.06364592e-01  5.68856638e-01  1.27080897e+02]
 [-1.02176144e-03 -8.13376831e-05  1.00000000e+00]]
```

4.

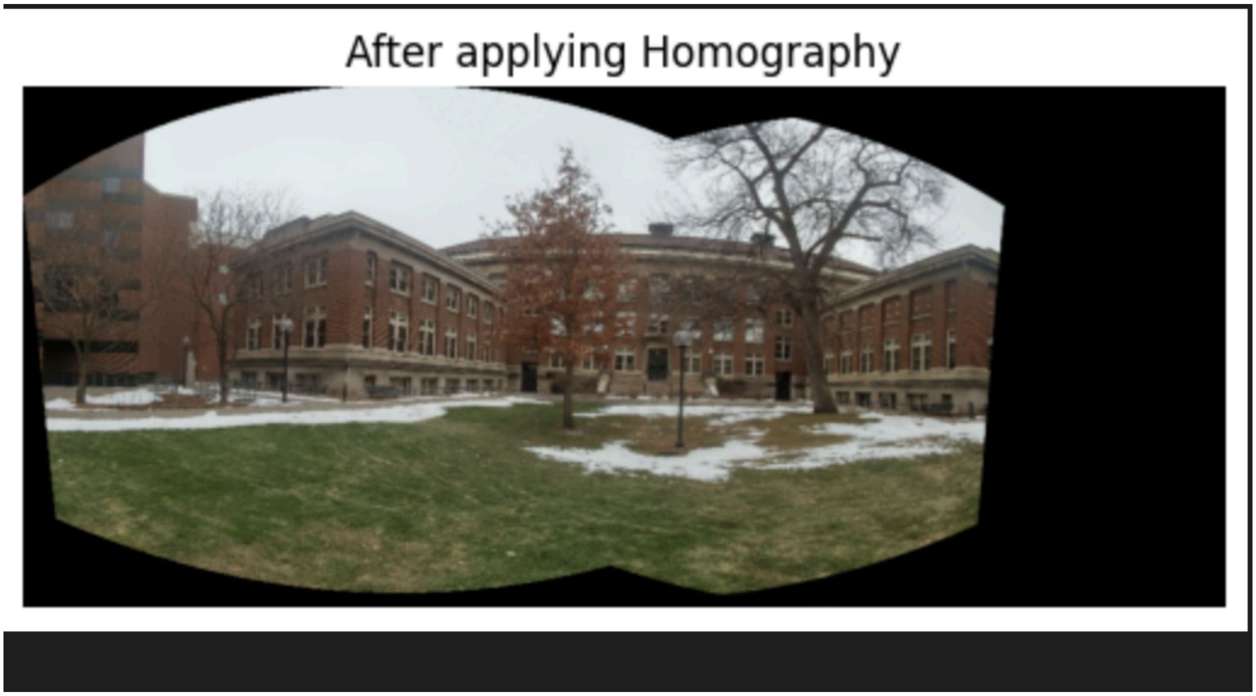
Then perspective warped the left and right view of the image using the snippet :

```
warped_l = cv2.warpPerspective(image1, H_R, (new_width_2, new_height_2))
warped_r = cv2.warpPerspective(image2, H_L, (new_width_1, new_height_1))
```

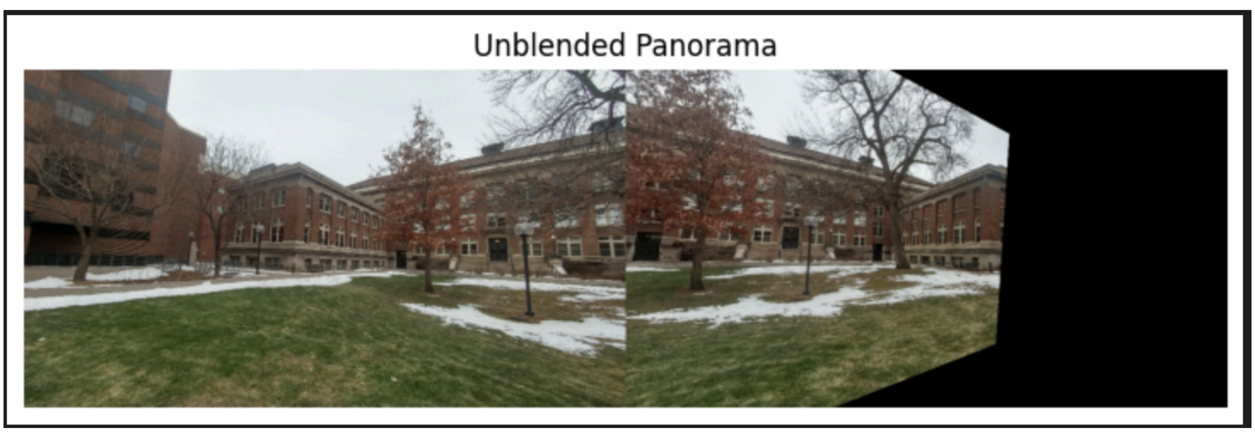
Then plotted the respective images side by side with the original one.



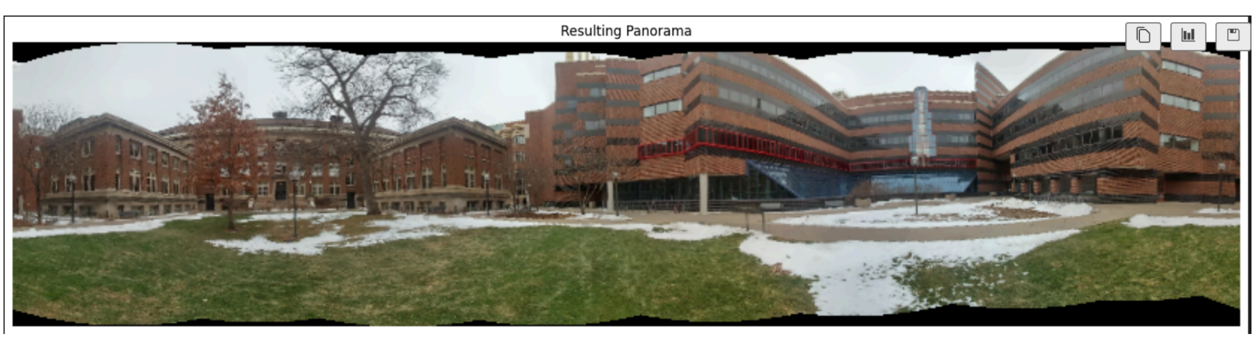
5.
Then stitched both the images using blending and one without blending :



Unblended one:



6.
Then stitched all the images in the “panaroma_generation” folder. Result:



Question 1 (Theory):

Aditya Arya
2021510

CV. Assign 3

1. $E = [t_x]R$ (Given)

Let e & e' be two epipoles.

∵ epipoles happen to be in null space

$$Ee = 0 \quad \text{--- (i)}$$
$$E^T e' = 0 \quad \text{--- (ii)}$$

Now,

$$[t_x]R \cdot e = 0.$$

∵ Rotational matrix is orthonormal, $RR^T = I$

Similarly, multiplying translational matrix $[t_x]$ with translational vector t, gives us 0.

$$[t_x] \cdot t = 0.$$

$$\therefore \quad e = R^T \cdot t$$

Shape $R = 3 \times 3$
$t = 3 \times 1$

We can verify that, $[t_x] \cdot R \cdot R^T \cdot t = 0$

↳ Associative property

Hence, $e = R^T \cdot t$.

Now, for second epipole,

$$E^T \cdot e' = 0.$$
$$\Rightarrow ([t_x]R)^T \cdot e' = 0$$
$$\Rightarrow R^T \cdot [t_x]^T \cdot e' = 0.$$
$$\Rightarrow -R^T \cdot [t_x] \cdot e' = 0 \quad [\because [t_x] \text{ is a skew symmetric matrix}].$$

We know that,

$[t_x] \cdot e' = 0$, will only be epipole if it is in the same direct as translational vector. (Cross product is 0).

$\therefore$ e' is scaled translational vector

$\Rightarrow e' = \lambda t$, (λ is a scalar)

2. Given $R = I$ & $t = [t_x \; 0 \; 0]^T$

$$[t_x] = \begin{bmatrix} 0 & -t_z & t_y \\ t_x & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$\because t = [t_x \; 0 \; 0]^T$

$$[t_x] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

$\therefore E = [t_x]R = [t_x] \cdot I = [t_x]$.

Let $x' = [x' \; y' \; 1]^T$

$x = [x \; y \; 1]$

$\therefore$

$$x'^T E x = 0 \Rightarrow [x' \; y' \; 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$= [x' \; y' \; 1] \begin{bmatrix} 0 \\ -t_x \\ t_y \end{bmatrix} = 0 \Rightarrow 0 - y' t_x + y t_x = 0$$

$\Rightarrow t_x y = y' t_x$

$\Rightarrow y = y'$ ✓