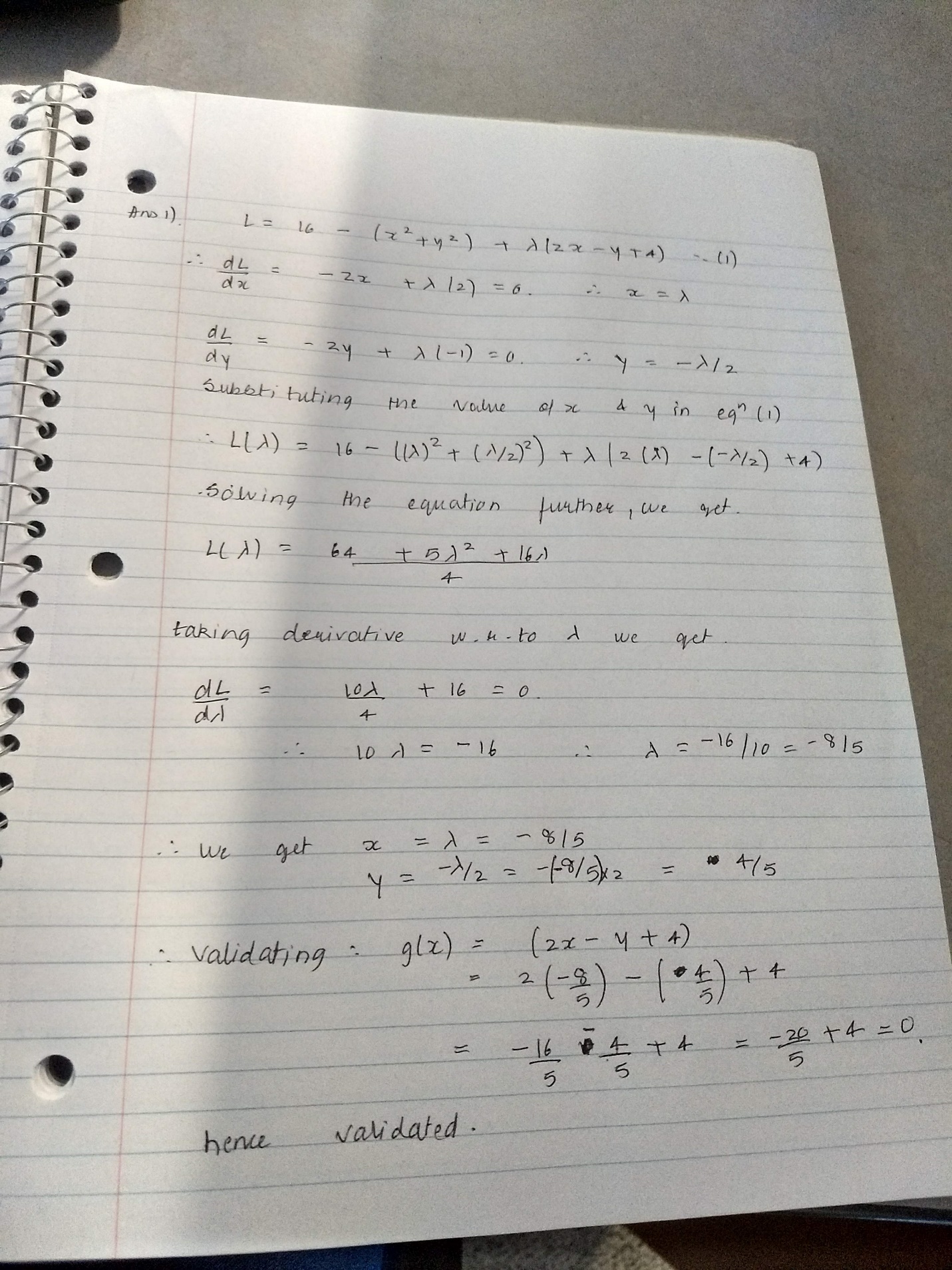
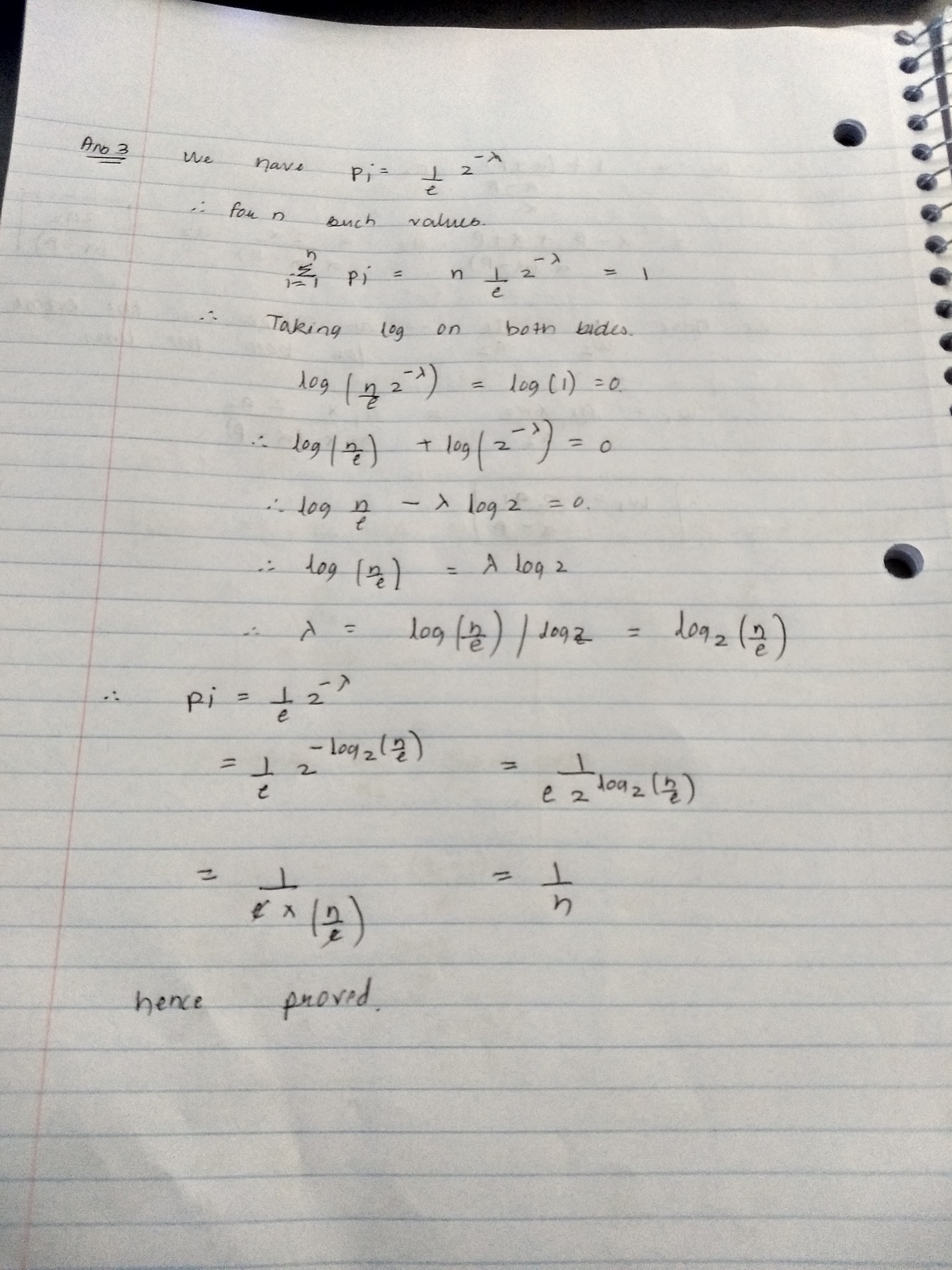
***Homework 5***

***Nikhil Lahoti***

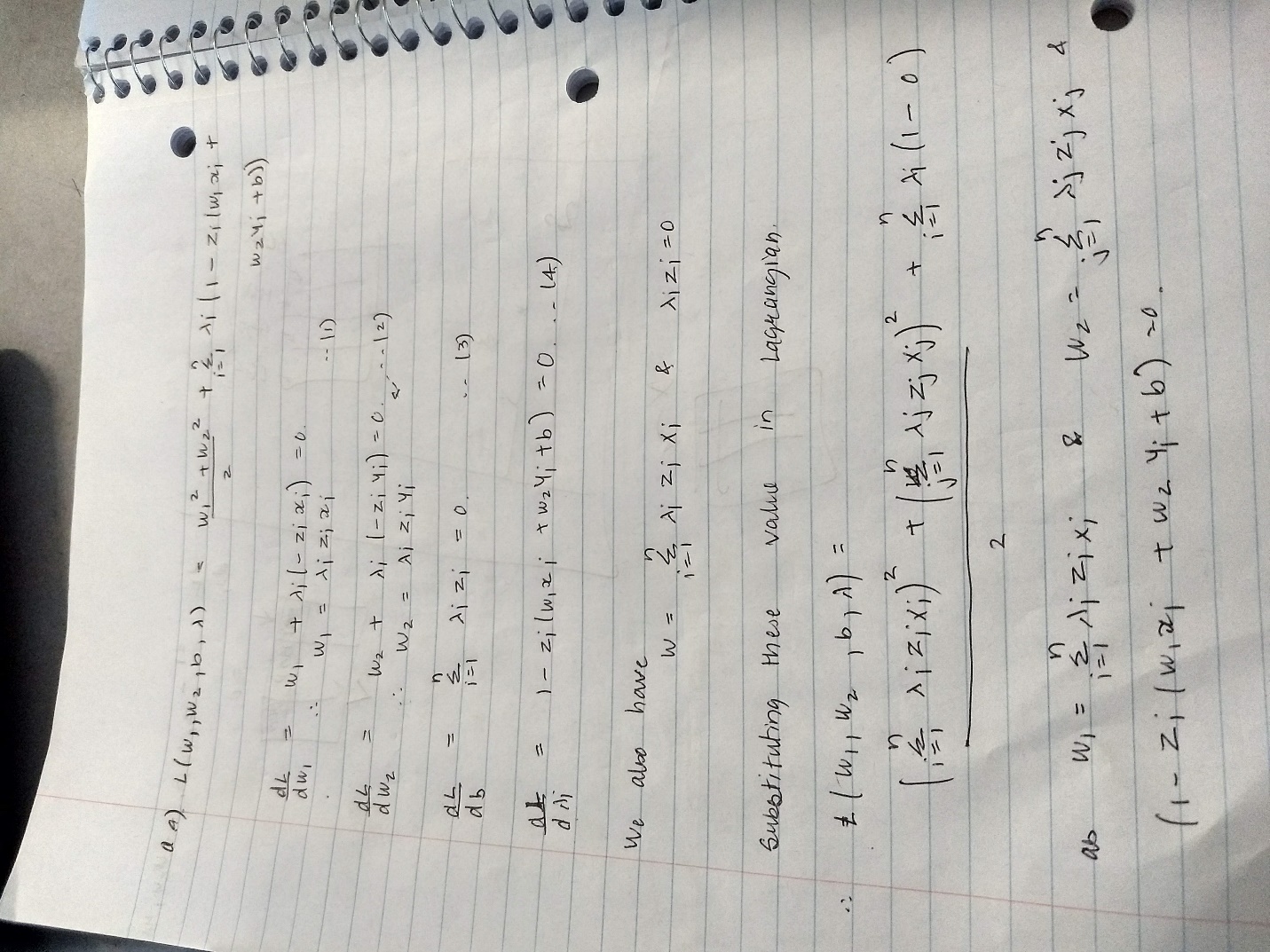
***012448514***

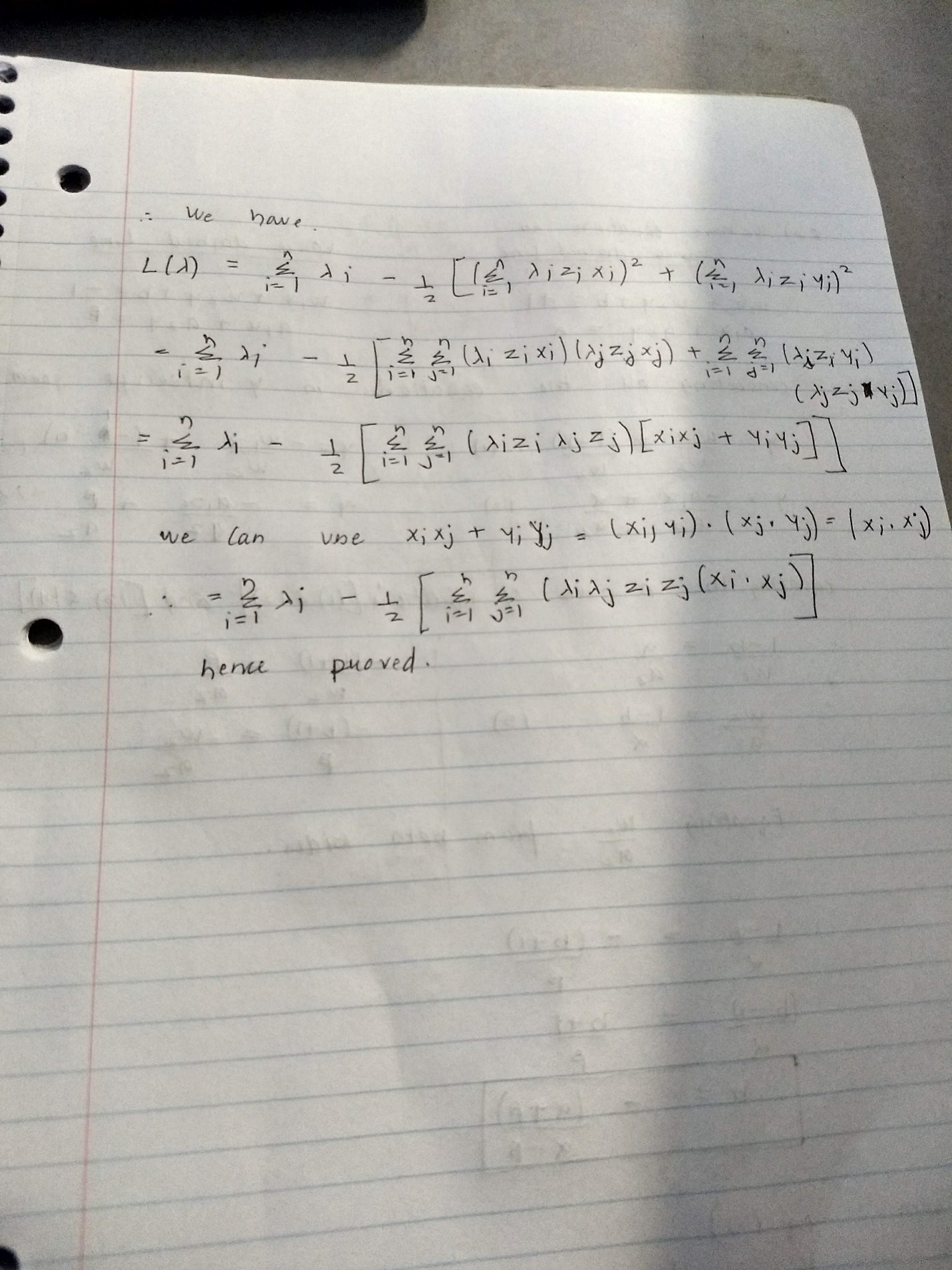
***Answer 1)***

**Answer 3)**

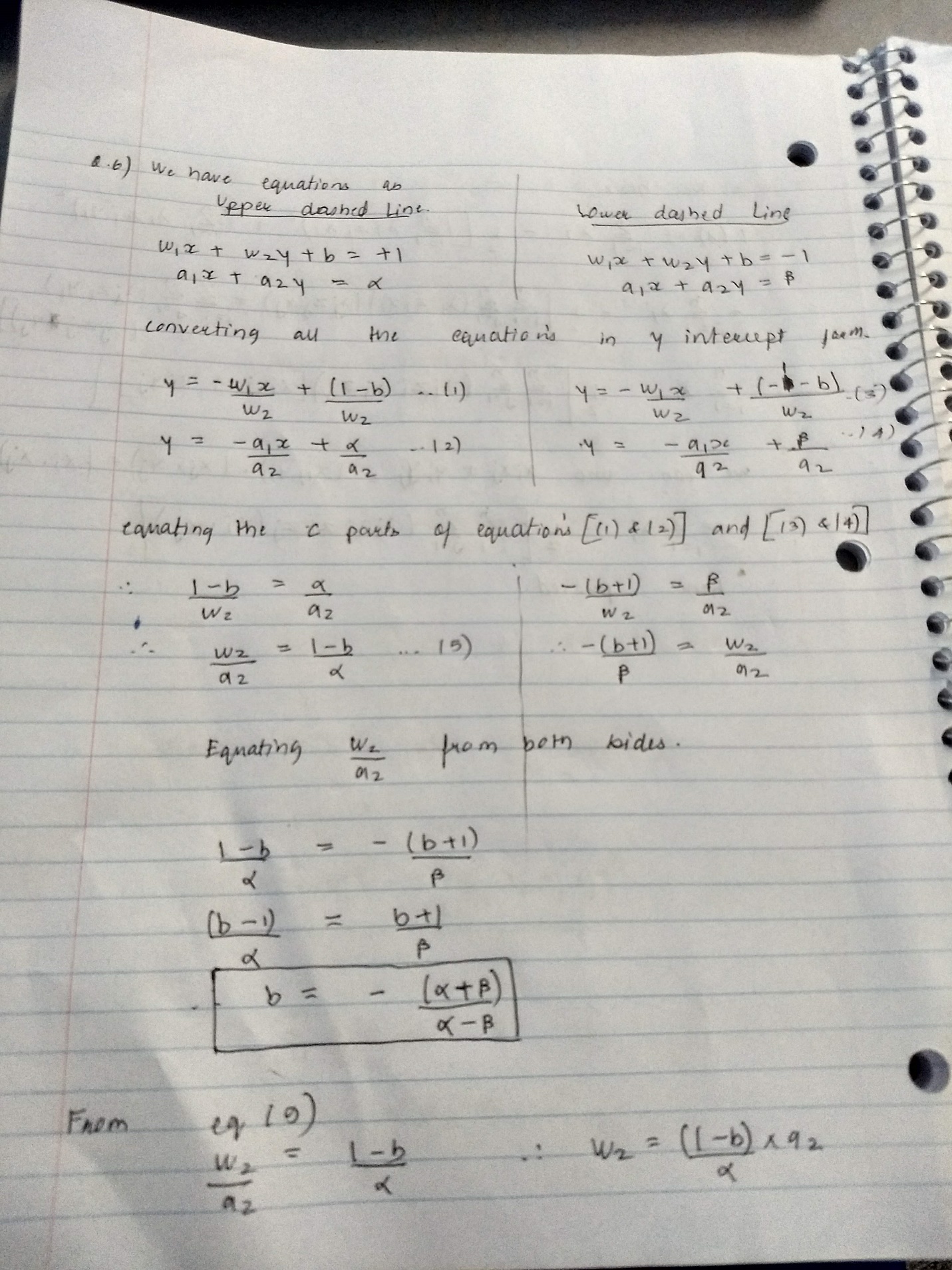


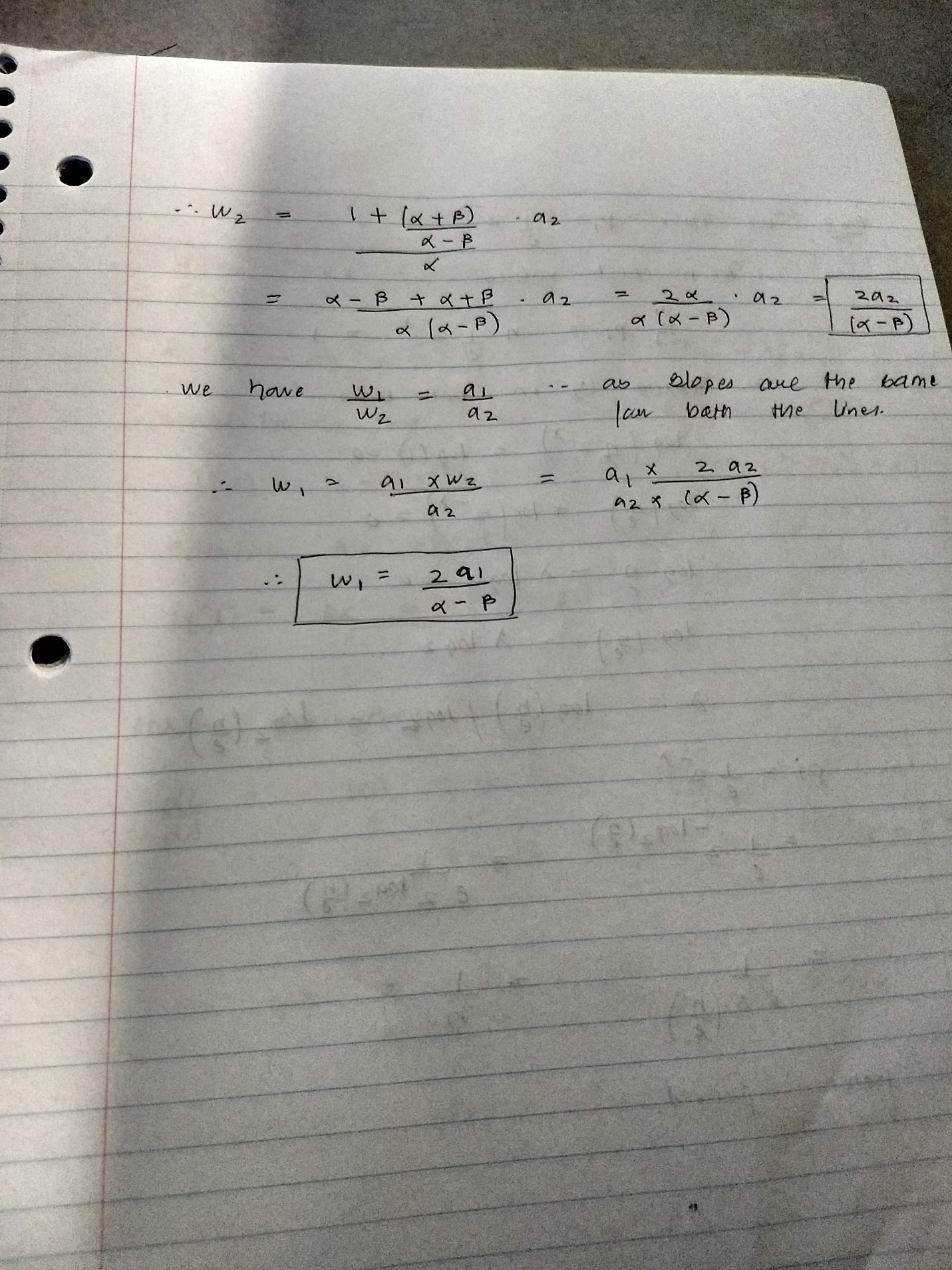
**Answer 4)**



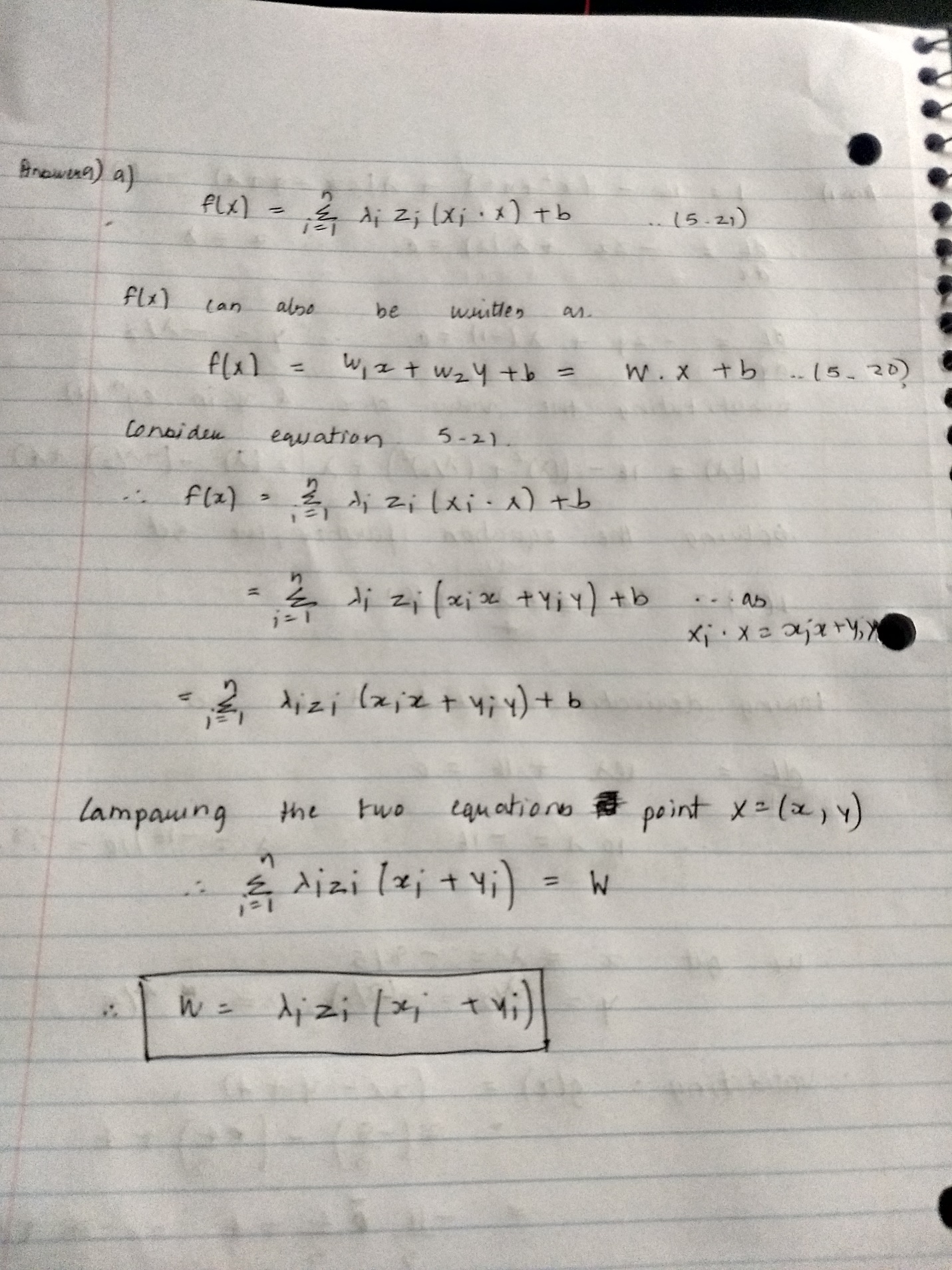


**Answer 6)**





**Answer 9) a)**



b) We can train the SVM to determine the weights of each feature. The weights tell the importance of each feature and hence more the weight, important is the feature. This way, we can ignore the features which have lower weight as they are not important as compared to others and can achieve dimensionality reduction.

c) In case of PCA, we get the importance in terms of Principal Components which is the linear combination of the features. We do not get the direct values of importance (weights) of the features and so it is difficult to say which features are important. Thus, it is difficult to achieve dimensionality reduction in terms of features (not in Principal Component terms).

In case of SVM, we get the weights (importance) of each feature and so it is easy to discard features based on the weights, as lower weights correspond to less importance.

In case of PCA, the eigen vectors and the eigen values can be determined easily from the Covariance matrix (as higher the values, higher is the variance, important is the feature) , and hence no training is required. But in case of SVM, the weights of the model are determined after performing the training. This is the disadvantage of SVM.

***Answer 12)*** Code is in the file: ***question12.py***

1. The accuracy obtained is 94.87 % (0.948717). The respective weights of HMM, SSD, OGS are

[[ 0.19688024 -0.53494025 -0.70487142]]

1. Here is the output:

With all 3 features

0.9487179487179487

Respective Weights of HMM, SSD, OGS

[[ 0.19688024 -0.53494025 -0.70487142]]

Accuracy after removing the HMM feature ->

0.9487179487179487

Respective Weights of SSD, OGS

[[-2.44489465 -2.24186897]]

Accuracy after removing the OGS feature ->

0.9487179487179487

Weight of SSD

[[-3.87596913]]

***Answer 15)***

a) Please find the code in **Question15a.py**

**Sample Output**

[0. 0. 2.5 0. 1.25 1.25]

-5.999999999999998

F[0] = 1.4999999999999982

z[0] = 1

F[1] = 2.7499999999999982

z[1] = 1

F[2] = 0.2500000000000018

z[2] = 1

F[3] = -3.499999999999999

z[3] = -1

F[4] = -1.0

z[4] = -1

F[5] = -1.0

z[5] = -1

b) Code in file **Question15b.py**

The values vary as the value of j is chosen randomly, here is one such instance

[ 0.22347138 0. 2.5 -0.54746154 2.67600985 0.59492308]

-4.7040393846154025

1. Please find the code in Question15c.ipynb

Sample output :

