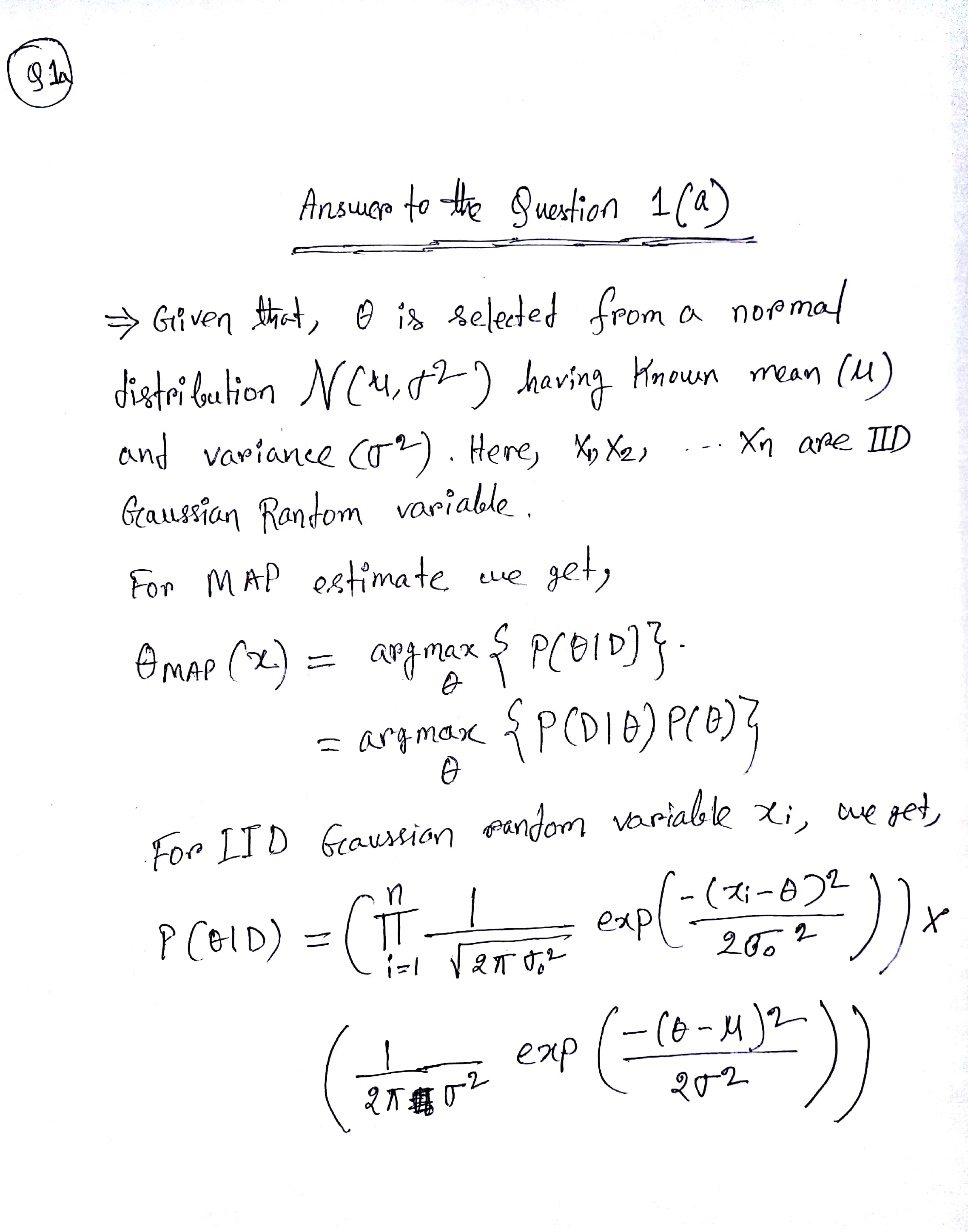
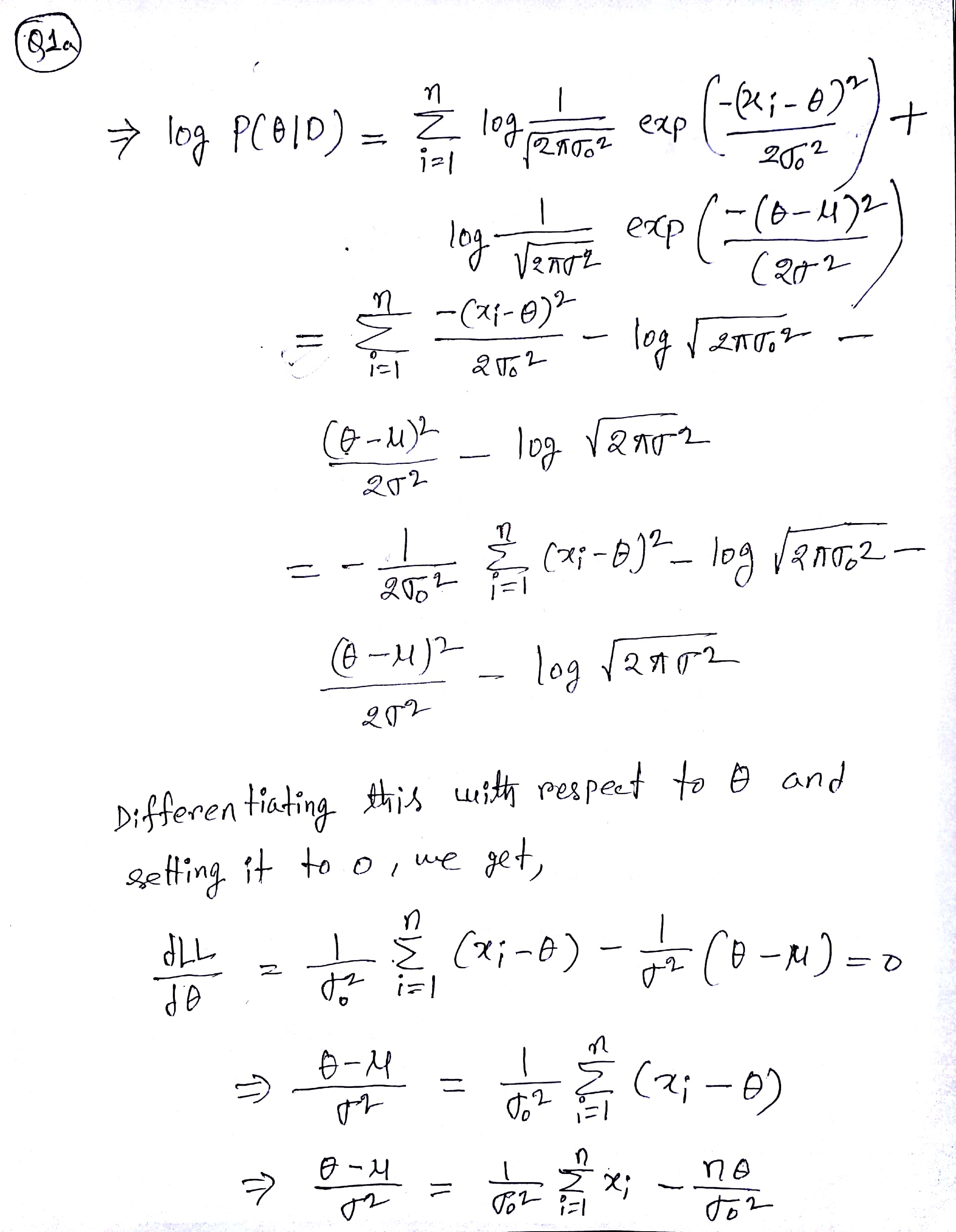
**CMPUT - 566**

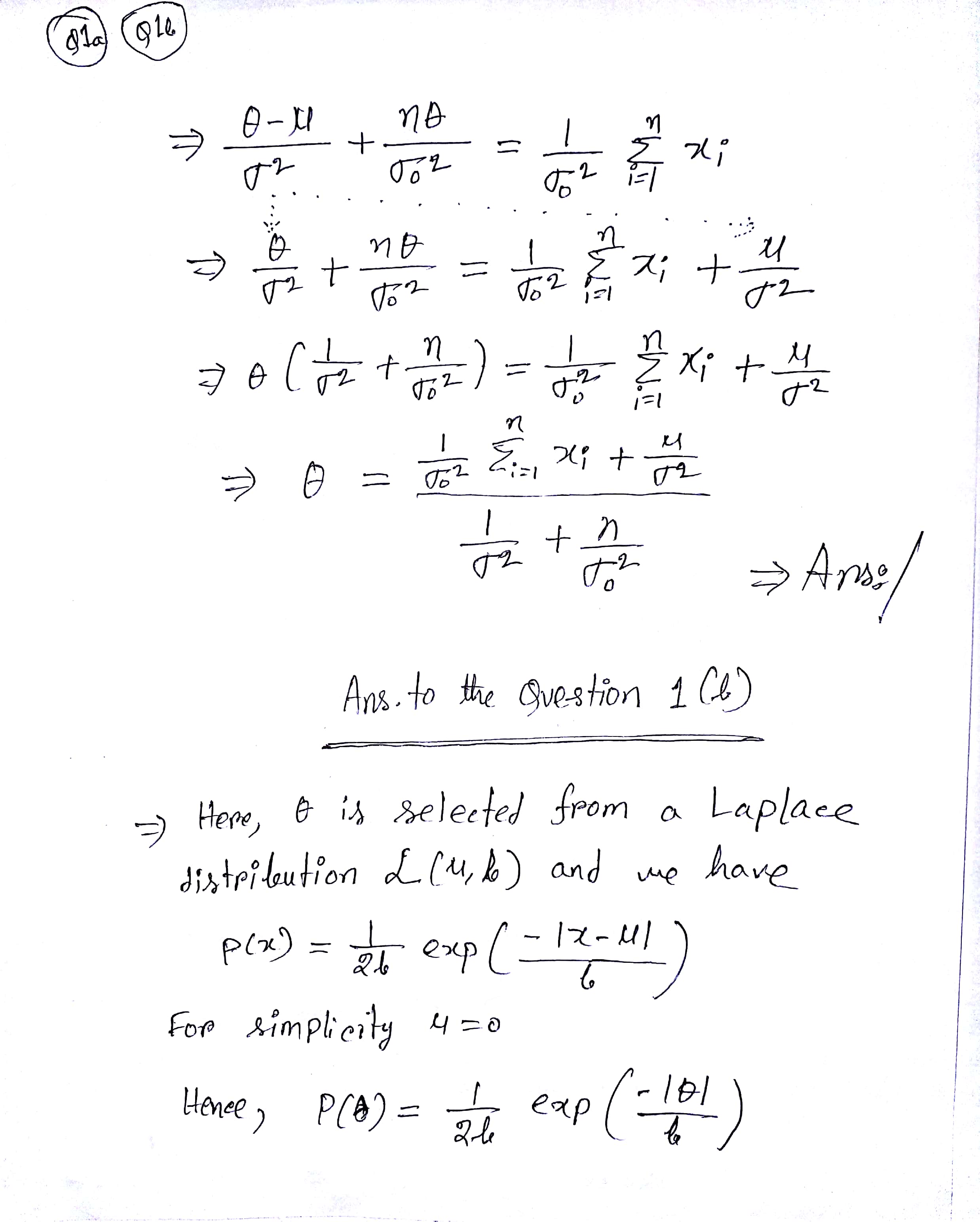
**Introduction to Machine Learning**

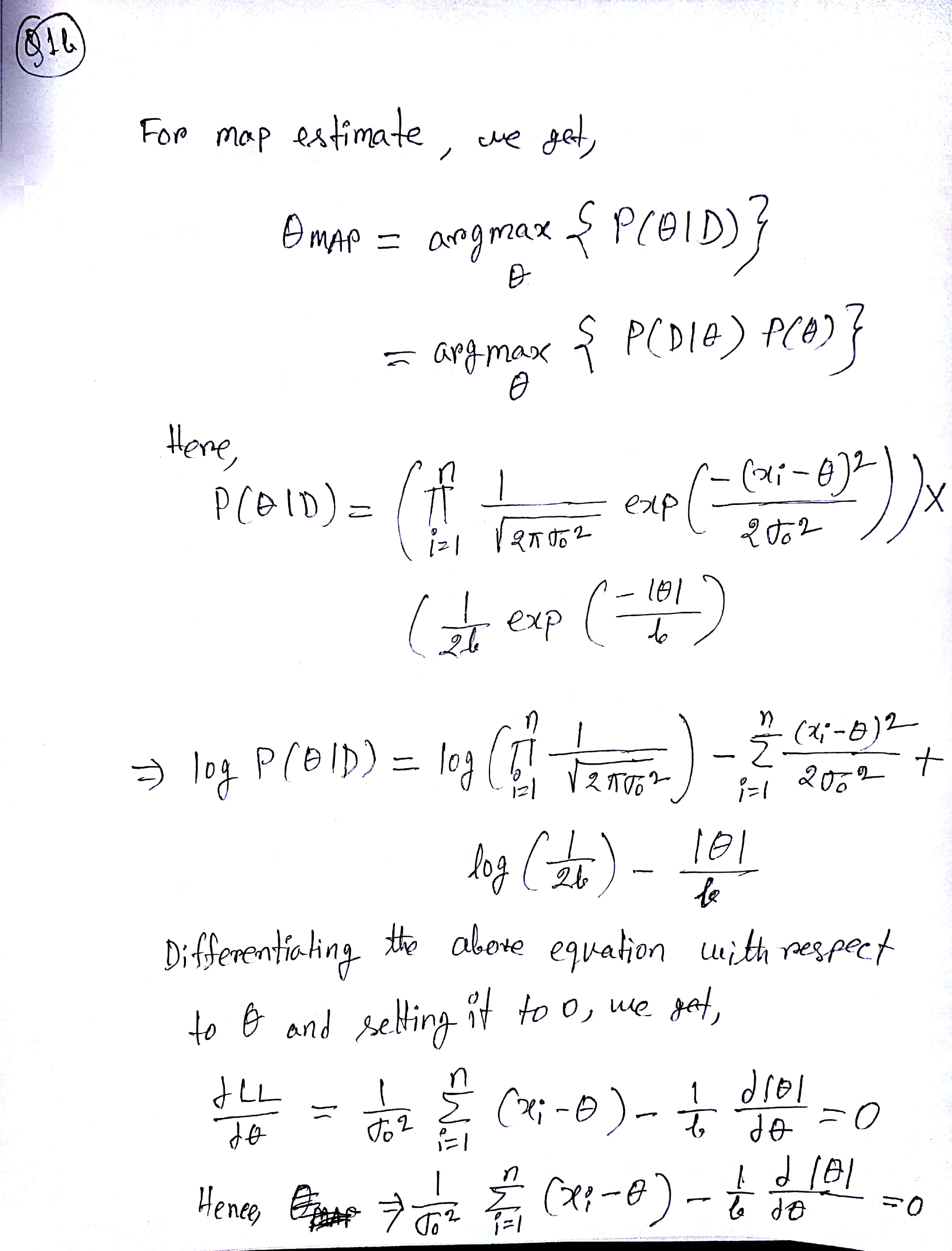
**Assignment – 2**

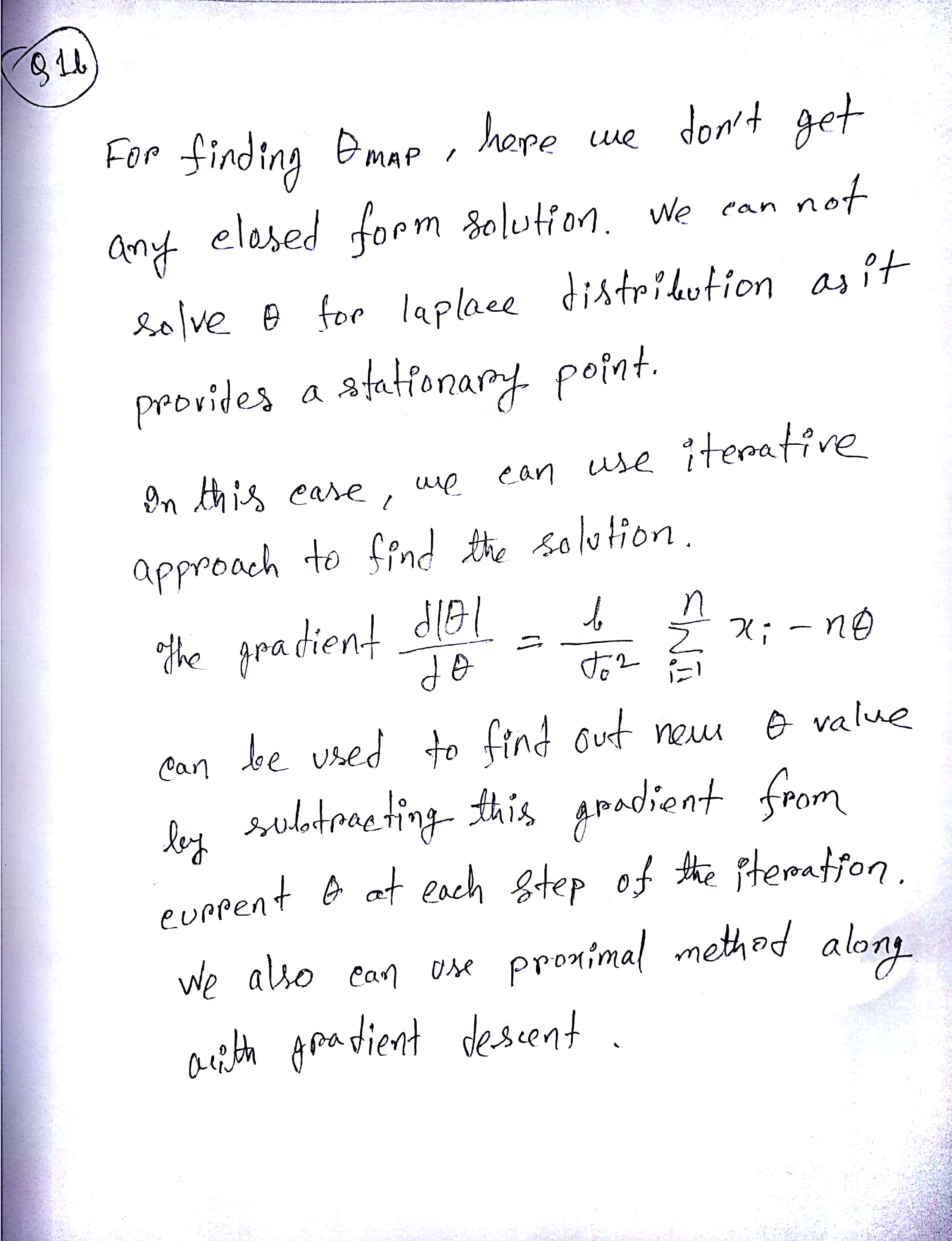
**Jakaria Rabbi**

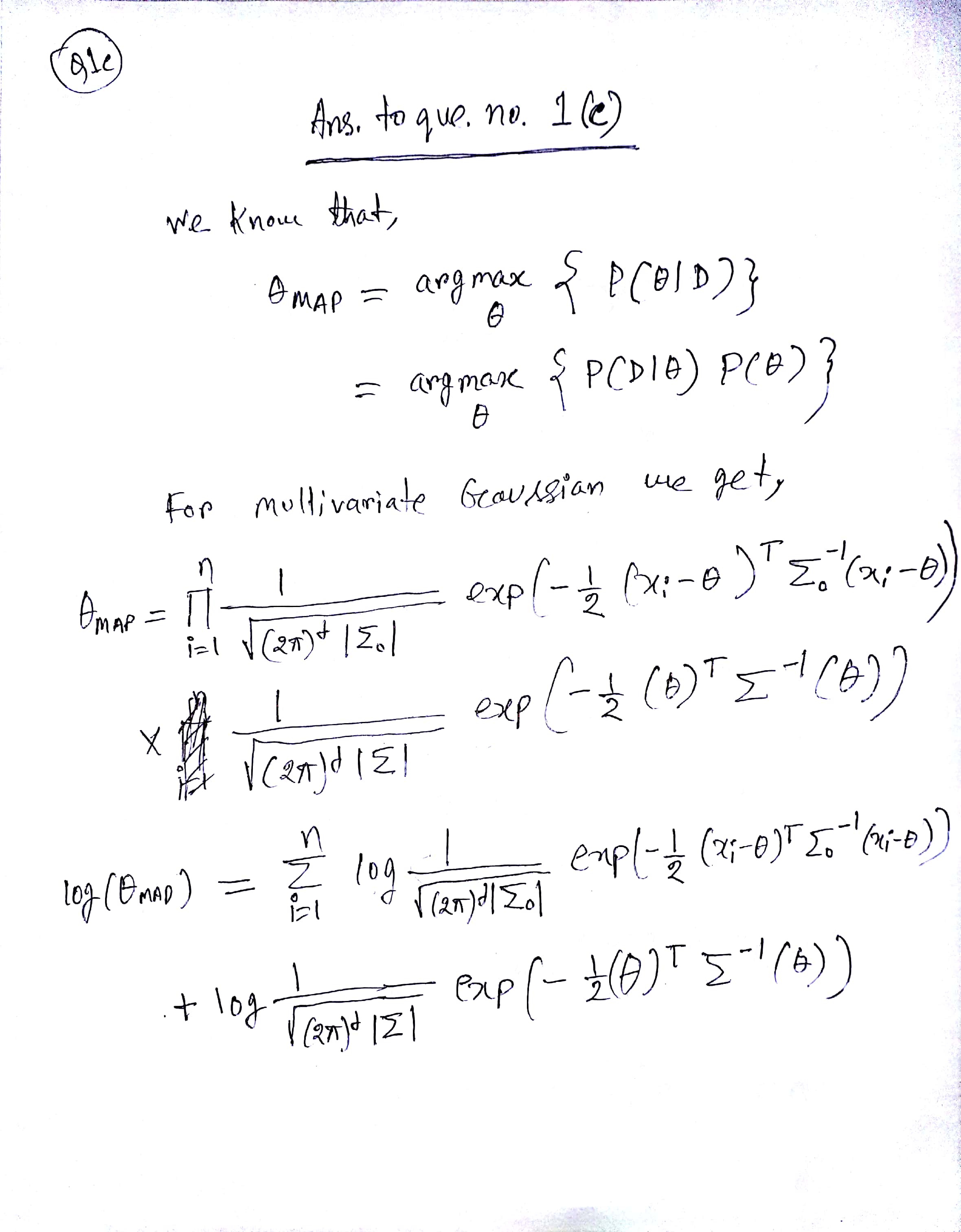


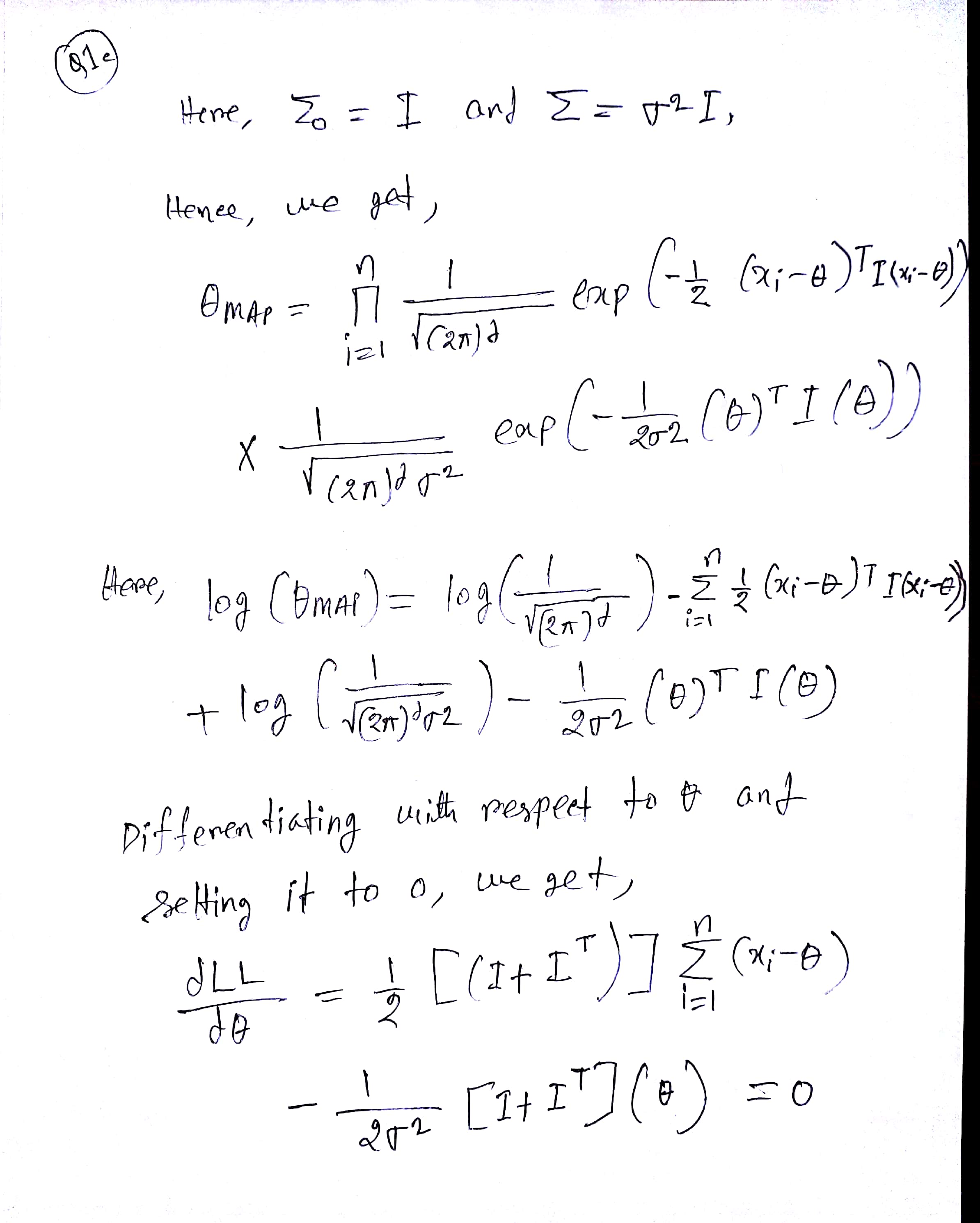


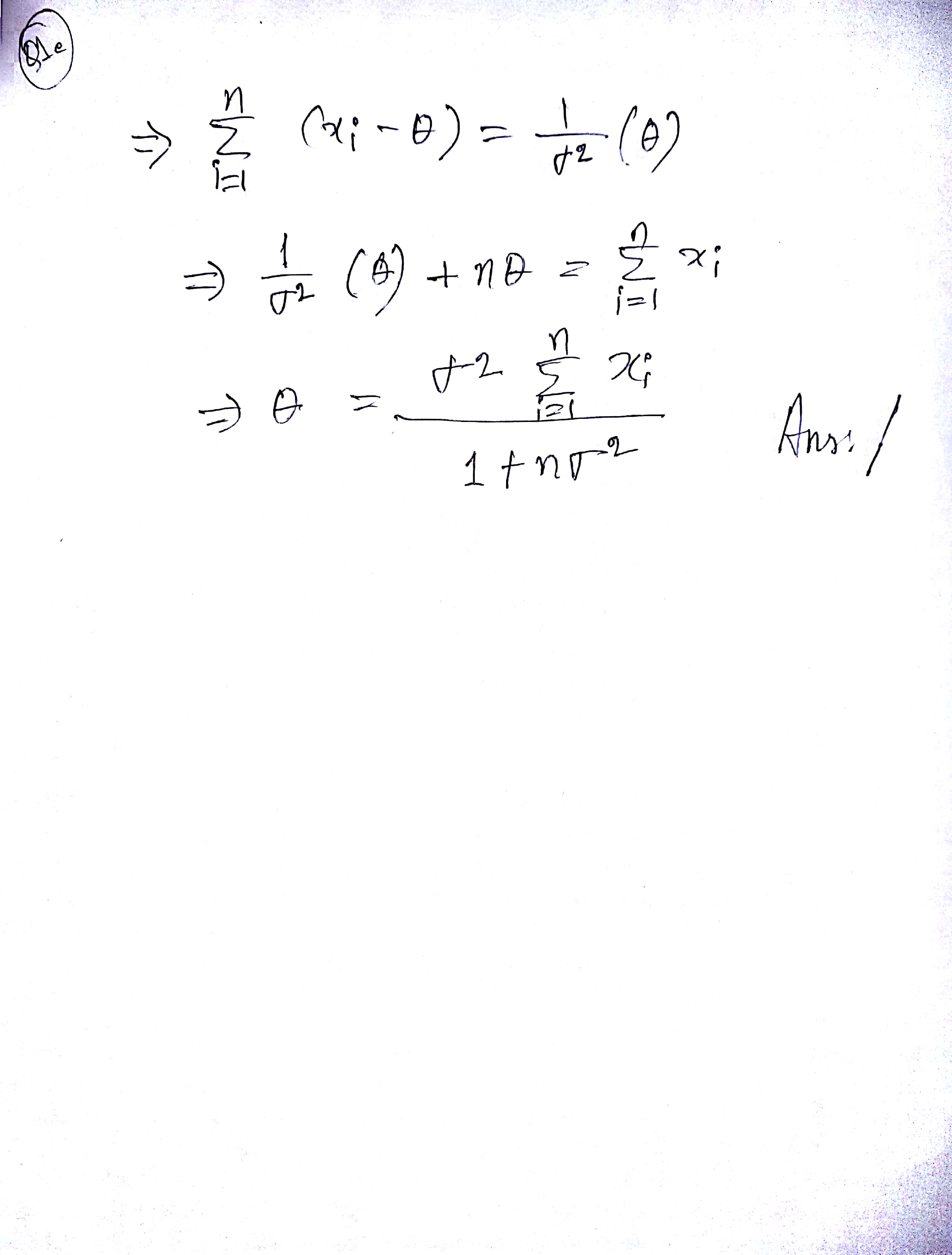












**Answer to the Question: 2(a)**

If we run the linear regression with full feature set, we get singular matrix error due to the singularity of (XTX). This error generally occurs when there are redundant features in the feature set and there is lack of linear independence between columns of the feature set. Also, if there is near linear dependence between the columns, this error generally occurs. The determinant of the matrix will become zero in that case. Here, we are trying to get a closed form solution and in that case, we need the calculate inverse of the feature matrix which is not possible for singular matrix.

We can easily solve this problem by using pseudo-inverse of the feature matrix. We can also add regularization term to the regression algorithm such as Laplace or Ridge regularizer, which will shift or truncate the small singular values which cause numerical stabilities. Another way is to use iterative approaches such as Batch or stochastic gradient descent.

**Answer to the Question: 2(b)**

Standard Error is implemented in the main method.

**Answer to the Question: 2(c)**

The Ridge regression is implemented as RidgeLinearRegression class inregressionalgorithms.py file.  
The results for regularization parameter λ = 0.01 are following:

1. Regularizer prevent the singular matrix error.
2. **Average Test Error: 41.77 Average Standard Error: 0.55**

**Answer to the Question: 2(d)**

The Lasso regression is implemented as LassoRegression class inregressionalgorithms.py file.

**Average Test Error: 42.46 Average Standard Error: 0.078**

**Answer to the Question: 2(e)**

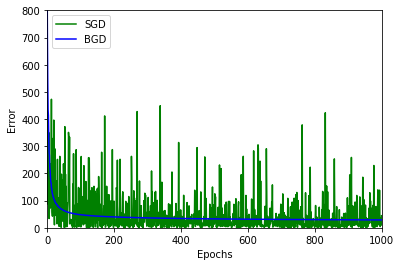
The Stochastic Gradient Descent is implemented as StochasticGradientDescent class inregressionalgorithms.py file. Process the whole data on 1000 times. After 1000 Epoch:

**Average Test Error: 42.82 Average Standard Error: 0.04**

**Answer to the Question: 2(f)**

The Batch Gradient Descent is implemented as BatchGradientDescent class inregressionalgorithms.py file. Process the whole data on average 2600 times.

**Average Test Error: 41.35 Average Standard Error: 0.31**



**Answer to the Question: Bonus(a)**

The Stochastic Gradient with RMSPROP is implemented as StochasticGradientWithRMSPROP class inregressionalgorithms.py file. After 1000 Epoch:

**Average Test Error: 47.94 Average Standard Error: 1.52**

**Answer to the Question: Bonus(b)**

The Stochastic Gradient with AMSGRAD is implemented as StochasticGradientWithAMSGRAD class inregressionalgorithms.py file. After 1000 Epoch:

**Average Test Error: 46.39 Average Standard Error: 1.4**