# EE2024 Assignment 1 Report

## Problem statement

Gradient descent algorithm is a way making use of the property of function gradient to find local minimum value for a given function and we are supposed to write the assembly language function optimize() in the file “optimize.s” to implement the gradient descent algorithm.

## Design methodology

1. Dealing with floating point numbers in assembly language is quite complicated. So casting a floating point number to integer value in C is necessary. However, simple casting may result in loss of precision and therefore we decide to preform scaling for the input of x.

So we multiply x by 100 and then convert x into an integer value in C and reasons for that are 1)we decide that it is enough for the precision to be 0.01 and 2)multiply x by 100 first and then casting will also reduce the loss of precision when perform conversion.

1. For a certain pair of a and b, the optimized solution (-b/(2a)) is not necessary an integer value. So if we scale up the optimized value and then there is better chance for us to get close to the value. For example, when the optimized value is 0.75, the function in assembly language can at best return 0 or 1; but if we scale 0.75 to be 75 and then we scale the result we get from assembly language back in C—we may get 74/100=0.74, which is more precise.
2. …

Of course the solution described above is not perfect—since we preform scaling, there may be the risk of overflow. Also that due to the limitation of the algorithm, the value of lambda is constrained by a, which will be covered in discussion.

## Implementation

## Result and discussion

## Extension

## Appendix