Data Structure

Project 1

Name: 이동민, 이민지

**Part1**

Matrix ADT

Objects: a list which have elements that same size of lists.

Functions:

* \_\_init\_\_(self: Matrix, x: int, y: int):
  + Initialize a matrix object which size is (x,y).
* Matrix read(text: str):
  + Read the text and create the Matrix(object) and return it
* Matrix read\_from\_file(filename: str):
  + Open the file, read the text and create Matrix.
* Matrix identity(n: int):
  + Create identity matrix of size n by n and return it
* Matrix add(self: Matrix, other: Matrix):
  + if self size and other matrix’s size are equal, add two matrices and return it.
  + if not, raise error.
* Matrix mul(self: Matrix, other: Matrix):
  + If self col size and other matrix’s row size are equal, multiply two matrices and return it.
  + If not, raise error.
* Matrix transpose(self: Matrix):
  + Transpose the self and return it
* Boolean symmetric(self):
  + If self is symmetric, return True
  + If not return False
* Boolean eq(self: Matrix, other: Matrix):
  + If self and other are same, return True
  + If not return False

**Part2**

(1)

Attached project code and commented on source code.

(2)

-add: O(mn) (calculating the m x n matrix)

-mul: O(mnl) (calculating the m x n matrix and n x l matrix)

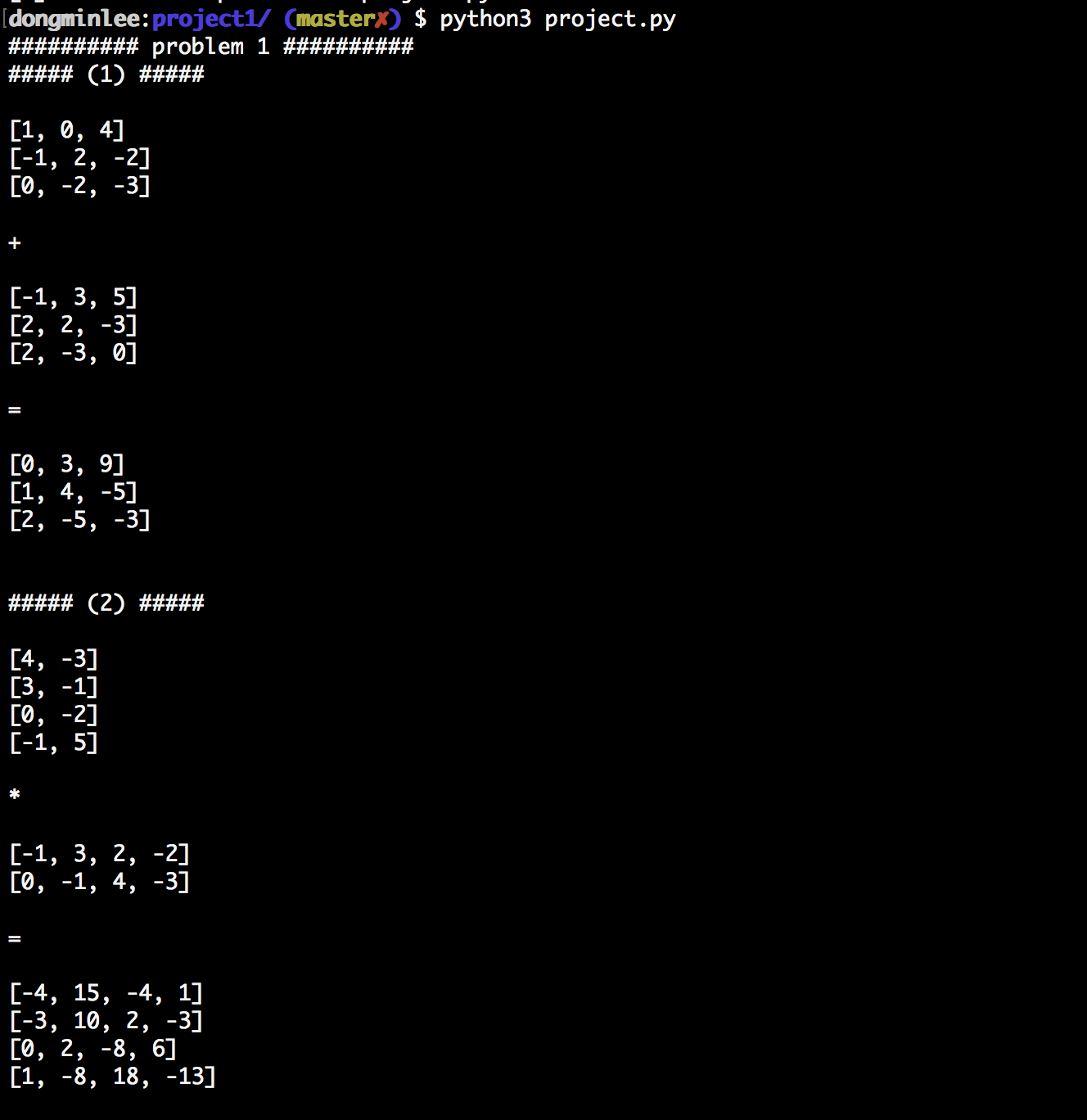
-transpose: O(mn) (calculating the m x n matrix)

-symmetric: O(mn) (calculating the m x n matrix)

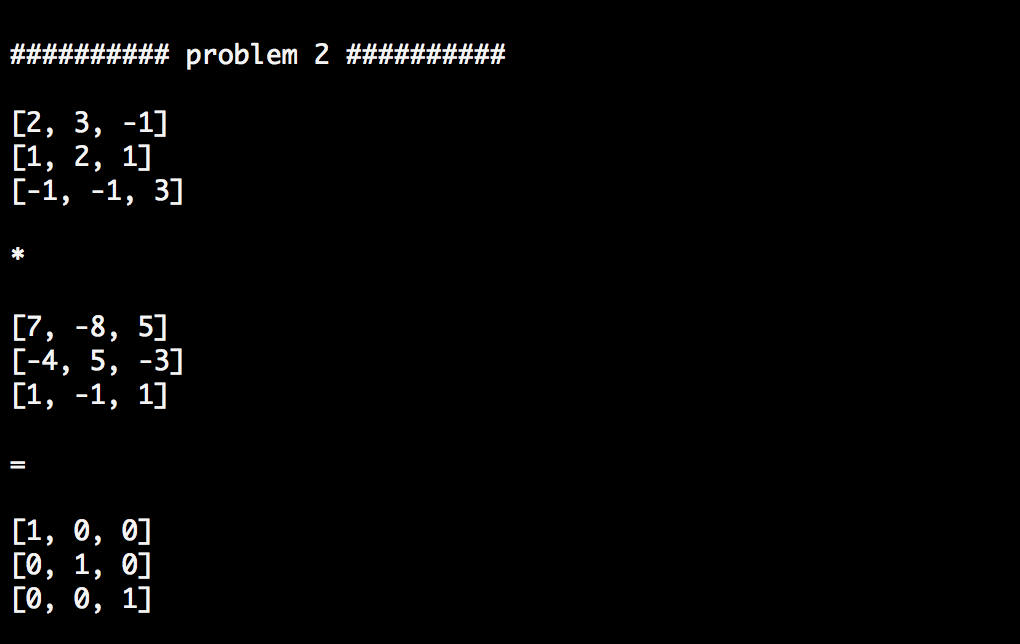
-eq: O(mn) (calculating the m x n matrix and m x n matrix)

**Part3**

**Problem1**



**Problem2**



Since the result of the multiplication of the two matrices is an identity matrix, two matrices are in relation of inverse.

**Problem3**

**(1)**

Attached project code

**(2)** 