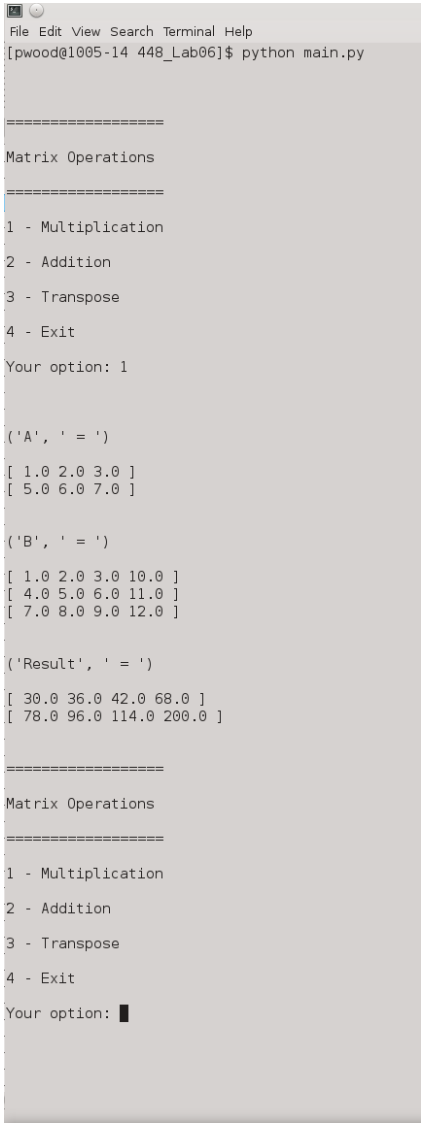
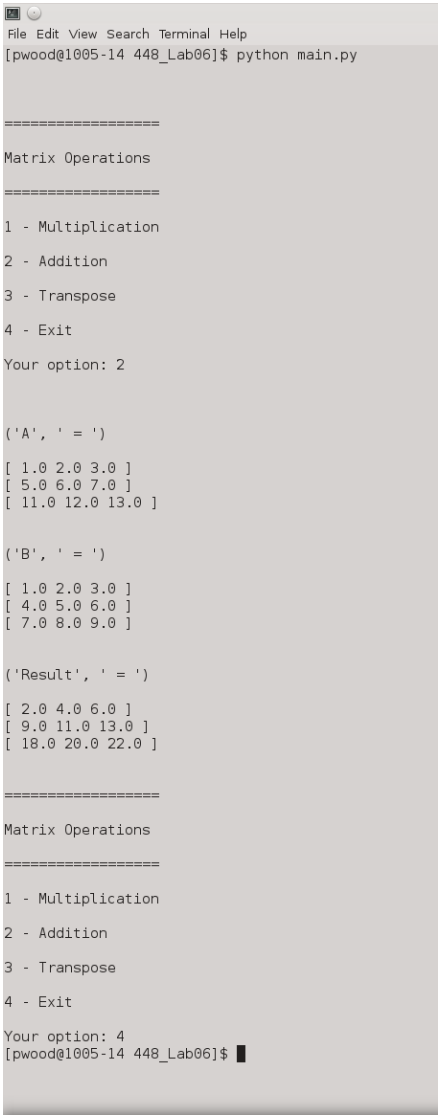
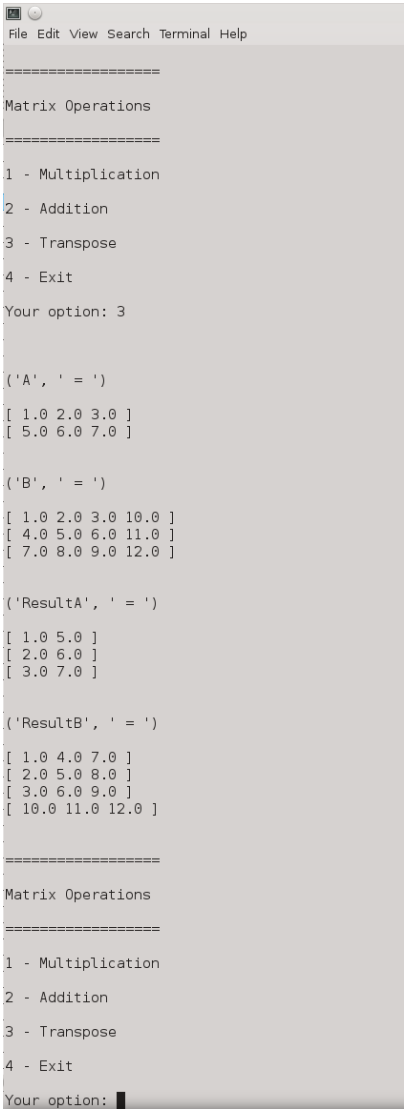
Philip Wood and Gustavo Nunes



Multiplication example



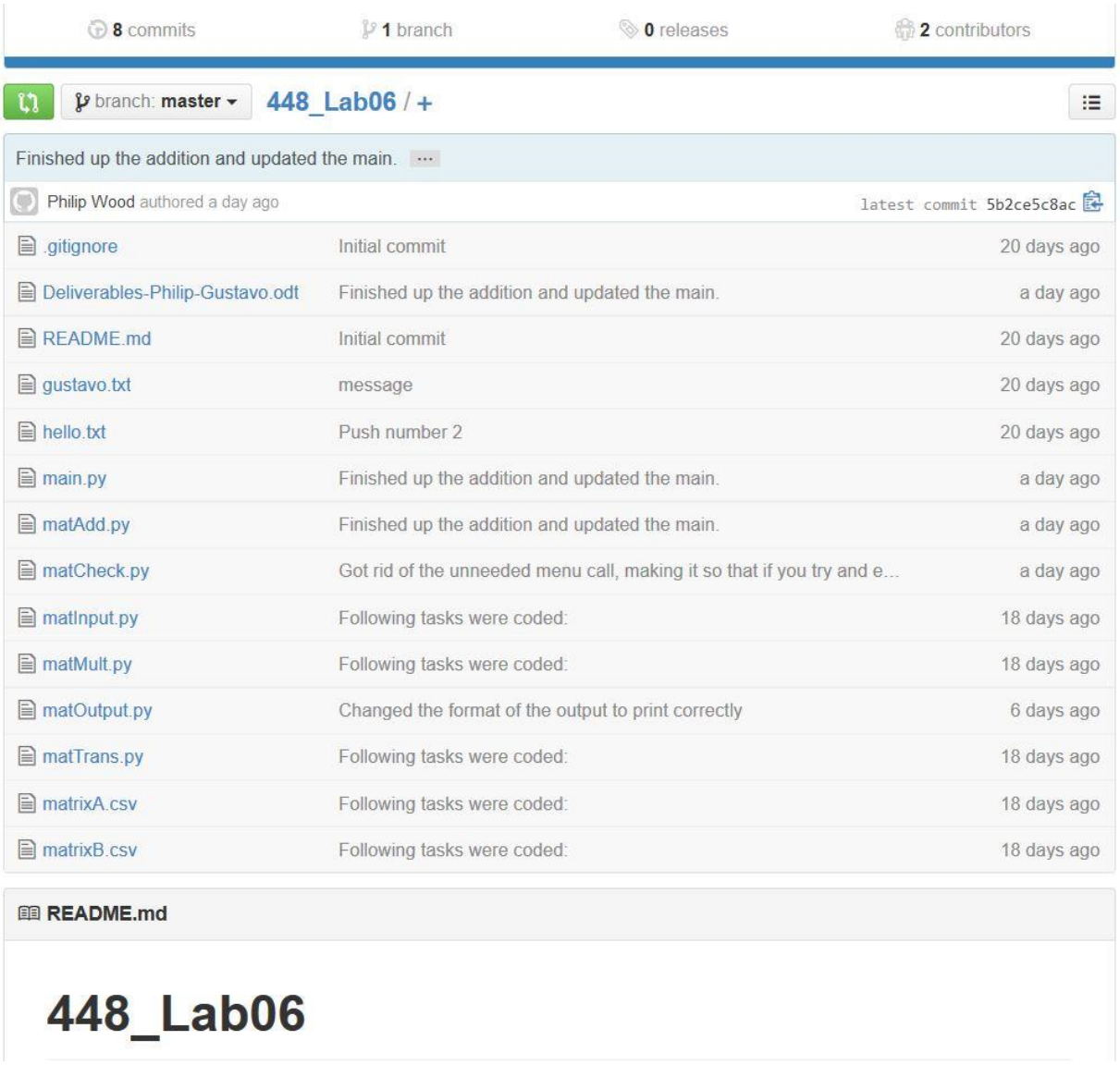
Addition example



Transpose example

Philip Wood and Gustavo Nunes

https://github.com/p3719/448_Lab06.git



Philip Wood and Gustavo Nunes

Main.py

```
import matInput
import matMult
import matTrans
import matOutput
import matCheck
import matAdd

def main():

    # Get matrices
    matrixA = matInput.getMatrixFromCSV('matrixA.csv')
    if not matCheck.CheckMatrix(matrixA):
            print("MatrixA is not a matrix!")
    matrixB = matInput.getMatrixFromCSV('matrixB.csv')
    if not matCheck.CheckMatrix(matrixA):
            print("MatrixB is not a matrix!")


    exit = False
    while(not exit):
        exit = menu(matrixA, matrixB)



def menu(matrixA = [], matrixB = []):
    # Ask operation
    print("\n\n")
    print("==================\n")
    print("Matrix Operations\n")
    print("==================\n")
    print("1 - Multiplication\n")
    print("2 - Addition\n")
    print("3 - Transpose\n")
    print("4 - Exit\n")
    operation = int(input('Your option: '))

    # Get matrices
    if not matrixA:
        matrixA = matInput.getMatrixFromCSV('matrixA.csv')
    if not matrixB:
        matrixB = matInput.getMatrixFromCSV('matrixB.csv')

    if operation != 4:
        matOutput.showMatrix(matrixA, 'A')
        matOutput.showMatrix(matrixB, 'B')

    # Choose the operation
```

```
    if operation == 1:
        if matrixA and matrixB:
            result = matMult.multiplication(matrixA, matrixB)
            if not result:
                print("\n\nMatrices cannot be multiplied.\n\n")
            else:
                matOutput.showMatrix(result, 'Result')
        else:
            print("\n\nMatrices were not found.\n\n")
    elif operation == 2:
        if matrixA and matrixB:
            result = []
            result = matAdd.addition(matrixA, matrixB)
            if not result:
                print("\n\nMatrices cannot be summed.\n\n")
            else:
                matOutput.showMatrix(result, 'Result')
        else:
            print("\n\nMatrices were not found.\n\n")
    elif operation == 3:
        if matrixA:
            resultA = matTrans.transpose(matrixA)
            matOutput.showMatrix(resultA, 'ResultA')
            resultB = matTrans.transpose(matrixB)
            matOutput.showMatrix(resultB, 'ResultB')

        else:
            print("\n\nMatrix was not found.\n\n")
    elif operation == 4:
        return True
    else:
        print("\n\nInvalid operation.\n\n")
    return False

main()
```

Philip Wood and Gustavo Nunes

matAdd.py

```
def addition(matrixA, matrixB):
	matrixC = []
	#  If the matrixes can be added then they are
	if CheckAdditionSize(matrixA, matrixB):
		for rows in range(len(matrixA)):
			matrixC.append([])
			for cols in range(len(matrixA[0])):
				matrixC[rows].append(matrixA[rows][cols] + matrixB[rows][cols])
	return matrixC

def CheckAdditionSize(matrixA, matrixB):
	#  If the matrixes are of the same size return true otherwise false
	if len(matrixA) == len(matrixB) and len(matrixA[0]) == len(matrixB[0]):
		return True
	return False
```

matCheck.py

```
def CheckMatrix(matrix):
	for x in range(len(matrix)):
		if x == 0:
			cols = len(matrix[x])
		else:
			if len(matrix[x]) != cols:
				return False
	return True
```

Philip Wood and Gustavo Nunes

matInput.py

```
def getMatrixFromCSV(fileName = 'matrixA.csv'):
    file = open(fileName)

    rowsSize = []
    csvFormat = True
    matrix = []
    for line in file:
        buffer = line.replace(" ", '').replace("\n", '')
        if buffer:
            buffer = list(buffer.split(','))
            auxBuffer = []
            for num in buffer:
                auxBuffer.append(float(num))
            buffer = auxBuffer
            if not rowsSize:
                rowsSize = len(buffer)
            if rowsSize != len(buffer):
                csvFormat = False
                break
            else:
                matrix.append(buffer)
    else:
        return matrix
```

Philip Wood and Gustavo Nunes

matMult.py

```
def multiplication(matrixA, matrixB):
    matrixC = []
    if checkMultiplicationSize(matrixA, matrixB):
        for i in range(len(matrixA)):
            matrixC.append([])
            for j in range(len(matrixB[0])):
                matrixC[i].append(0)

        for row in range(len(matrixA)):
            for column in range(len(matrixB[0])):
                matrixC[row][column] = 0
                for common in range(len(matrixB)):
                    matrixC[row][column] += matrixA[row][common] * matrixB[common][column]
    return matrixC

def checkMultiplicationSize(matrixA, matrixB):
    return len(matrixA[0]) == len(matrixB)
```

Philip Wood and Gustavo Nunes

matOutput.py

```
import sys

def showMatrix(matrix, label = ''):
    print("\n\n")
    if label:
        print(label, " = ")
    for row in range(len(matrix)):
        printf("\n[ ")
        for column in range(len(matrix[0])):
            printf(str(matrix[row][column]))
            printf(" ")
        printf("]")

def printf(message):
            sys.stdout.write(message)
```

matTrans.py

```
def transpose(matrixA):
    matrixB = []
    numRows = len(matrixA)
    numColumns = len(matrixA[0])
    for row in range(numColumns):
        matrixB.append([])
        for column in range(numRows):
            matrixB[row].append(0)
            matrixB[row][column] = matrixA[column][row]
    return matrixB
```