

```
#Jack Amos
#CS2300-002
#9/21/2019
#Project 1
#using Python 3.7
```

```
#func to write matrix A and B to files
def print_and_write(matrix):
```

```
    with open(matrix,'r') as file:
        contents = file.read()
```

```
    #get each value in string and convert it to integer
    matrix_values = [int(x) for x in contents.split()]
```

```
    #remove and save matrix dimensions
    matrix_row = matrix_values.pop(0)
    matrix_column = matrix_values.pop(0)
```

```
    row_start = 0
    row_end = matrix_column
    matrix_string = ""
```

```
    #sets limit so index won't go beyond array dimensions
    while row_start < matrix_row*matrix_column:
        #adds row of values to output string plus newline char
        matrix_string += str(matrix_values[row_start:row_end]) + "\n"
        #increment by column number so vars are at the start and end of next row
        row_start+=matrix_column
        row_end+=matrix_column
```

```
    #removes commas and brackets for beautification
    matrix_string = matrix_string.replace(",","").replace("[","").replace("]","")
    print(matrix_string)
```

```
    #appends matrix dimensions to front of output string
    matrix_string = str(matrix_row)+" "+str(matrix_column)+" "+matrix_string
```

```
    #sets file name
    file_name = "CS2300P1aAmos.out"+matrix[0]
```

```
    #writes output string to new file
    with open(file_name,"w+") as file:
        file.write(matrix_string)
```

```

#subtract matrix2 from matrix1
def subtract(matrix1,matrix2):

    with open(matrix1,'r') as file:
        contents = file.read()

    #get each value in string an convert it to integer
    matrix_values1 = [int(x) for x in contents.split()]

    #remove and save matrix dimensions
    matrix_row = matrix_values1.pop(0)
    matrix_column = matrix_values1.pop(0)

    with open(matrix2,'r') as file:
        contents = file.read()

    #get each value in string an convert it to integer
    matrix_values2 = [int(x) for x in contents.split()]

    #remove and save matrix dimensions
    matrix_row = matrix_values2.pop(0)
    matrix_column = matrix_values2.pop(0)

    index = 0
    matrix_sub = []

    #places subtracted value in new matrix
    for n in matrix_values1:
        matrix_sub.append(5*n - matrix_values2[index])
        index+=1

    row_start = 0
    row_end = matrix_column
    matrix_string = ""

    #sets limit so index wont go beyond array dimensions
    while row_start < matrix_row*matrix_column:
        #adds row of values to output string plus newline char
        matrix_string += str(matrix_sub[row_start:row_end]) + "\n"
        #increment by column number so vars are at the start and end of next row
        row_start+=matrix_column
        row_end+=matrix_column

    #removes commas and brackets for beautification
    matrix_string = matrix_string.replace(",","").replace("[","").replace("]", "")
    print(matrix_string)

    #appends matrix dimensions to front of output string
    matrix_string = str(matrix_row)+" "+str(matrix_column)+" "+matrix_string

```

```

#writes output string to new file
with open("CS2300P1aAmos.calc","w+") as file:
    file.write(matrix_string)

#transposes passed matrix
def transpose(matrix):

    with open(matrix,'r') as file:
        contents = file.read()

    #get each value in string and convert it to integer
    matrix_values = [int(x) for x in contents.split()]

    #remove and save matrix dimensions
    matrix_row = matrix_values.pop(0)
    matrix_column = matrix_values.pop(0)

    index = 0
    column_iterator = 1
    matrix_string = ""

    #matrix is "rolled over and flipped" when transposed, so number of rows now = number of
    #columns and vice versa
    while column_iterator <= matrix_row:
        #checks if index is within dimensions
        if index >= matrix_row*matrix_column:
            #moves index to next column, iterator to the next column and adds newline
            index = column_iterator
            column_iterator+=1
            matrix_string = matrix_string+"\n"
        #adds number to new matrix row and iterates down column
        else:
            matrix_string = matrix_string+" "+str(matrix_values[index])
            index+=matrix_column

    print(matrix_string)

    #appends matrix dimensions to front of output string
    matrix_string = str(matrix_row)+" "+str(matrix_column)+matrix_string

    #writes output string to new file
    with open("CS2300P1aAmos.trans","w+") as file:
        file.write(matrix_string)

```

```
#calling functions  
print_and_write("Amatrix")  
print_and_write("Bmatrix")  
subtract("Amatrix","Bmatrix")  
transpose("CS2300P1aAmos.calc")
```