Daniel Burger

Week 1- Programming Exercises

**Problem 1:**

################

# Author: Daniel Burger

# Date: 2/23/2019

# This program draws a patter of pairs of pound signs that have

# more spaec inbetween them the further down you go

#################

NUM\_LINES = 9 # number of lines the code will print

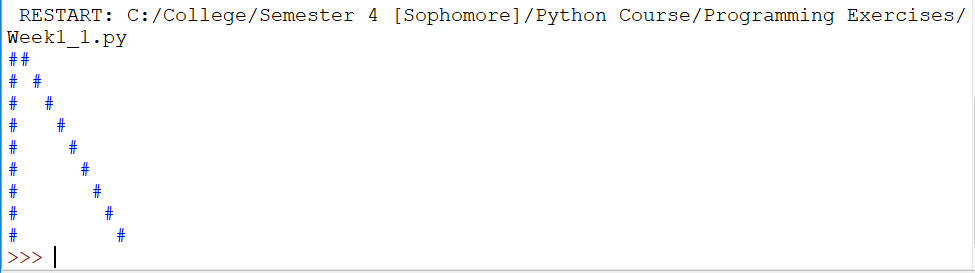
for i in range(NUM\_LINES):

print('#', end='') #prints the first pound sign, always first charcter each line

for j in range (i):

print(' ', end='') #prints as many spaces as whatever row it is in (starting with row 0)

print('#') #prints last pound sign, always last character on each line



**Problem 2:**

################

# Author: Daniel Burger

# Date: 2/23/2019

# This program finds and prints all prime numbers

# between and any given number (but in this case, 0 to 100)

#################

NUMBER = 100 #chosen number to print all prime numbers from 0 to NUMBER

for i in range(100):

#special cases for 2, 3, 5, and 7

if ((i == 2) | (i == 3) | (i == 5) | (i == 7)):

print(i)

if ((i % 2) != 0): #if i is not divisible by 2

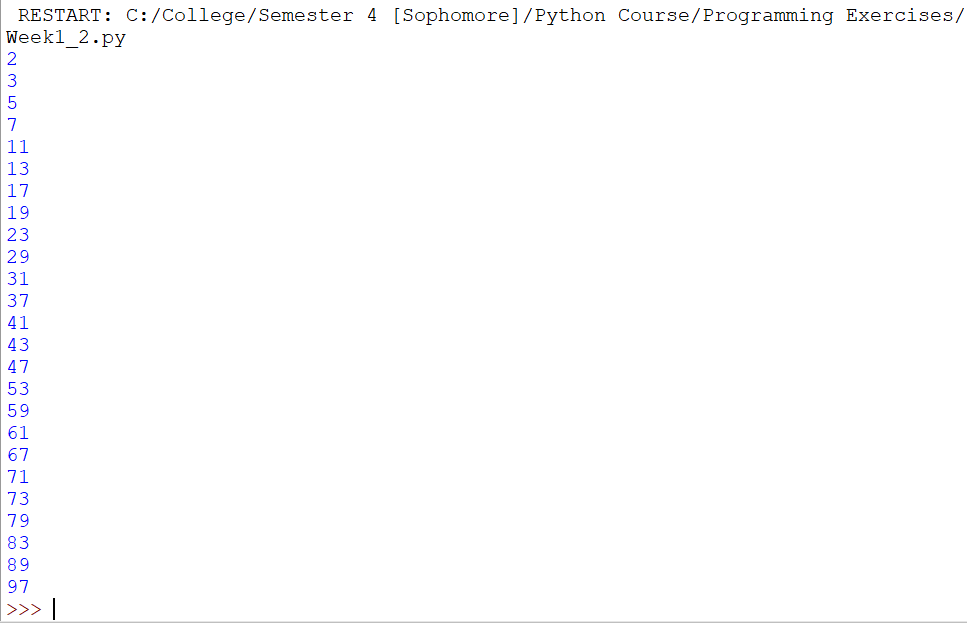
if ((i % 3) != 0): #if i is not divisible by 3

if ((i % 5) != 0): #if i is not divisible by 5

if ((i % 7) != 0): #if i is not divisible by 7

if (i != 1): #1 is not a prime number

print(i)



**Problem 3.a:**

################

# Author: Daniel Burger

# Date: 2/24/2019

# This program transposes matrix M and prints it

# in its proper transposed order

#################

M = [[2, -5, 8, 11],[3, 7, -9, -5], [4, 0, -1, 7]] #starting matrix

Mt = [] #transpose of M

for i in range(len(M[0])): #for every column

column = [] #create a new column from M that will b a row in Mt

for j in range(len(M)): #for every row

x = M[j][i] #pulls value from M

column.append(x) #appends value to column string

Mt.append(column) #appends column as a row

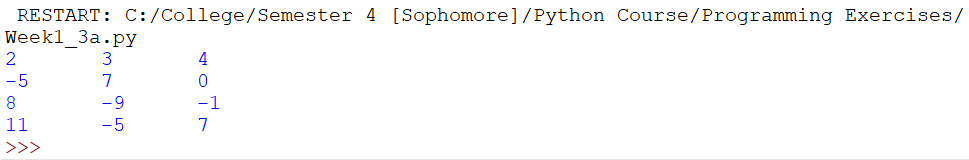
#prints the matrix in an easy to read format

for r in range(len(Mt)):

for c in range(len(Mt[0])):

print(str(Mt[r][c]) + '\t',end='')

print()



**Problem 3.b:**

################

# Author: Daniel Burger

# Date: 2/24/2019

# This program multiplies two matrices and prints

# results in matrix format

#################

M = [[2, -5, 8, 11],[3, 7, -9, -5], [4, 0, -1, 7]] #matrix

Mt = [[2, 3, 4], [-5, 7, 0], [8, -9, -1], [11, -5, 7]] #transposed matrix

product = [] #where the answer will be stored, in matrix format

for i in range(len(M)): #for every number of rows in M

for j in range(len(Mt[0])): #for every number of cols in Mt

temp\_row = [] #temporary row array to fill values and eventually append

for k in range(len(Mt)):

x = M[i][k]\*Mt[k][j] #calculation done for matrix multiplication

temp\_row.append(x)

product.append(temp\_row) #adding the product row to matrix 'product'

#prints the matrix in an easy to read format

for r in range(len(product)):

for c in range(len(product[0])):

print(str(product[r][c]) + '\t',end='')

print()

