CSE101 HOMEWORK#4

(Create folder named as your_name_hw4. Save your Python scripts as q1.py, q2.py and q3.py. Rar or zip the folder and submit it to Coadsys)

1) (30p) Using while loops write a Python program which generates a random number N and asks the user to guess this number. If user's guess is lower than N your program should print "Guess higher"; if it is higher it should print "Guess lower". When N is found by the user, your program should print the number of guesses user made so far to find the correct number. Random number generation is given below. Example output is also shown below.

- Random number generation: You should place the given code to the top of your program

```
import random
N=random.randint(0, 100)
```

- Example output:

```
Guess a number: 50
Guess lower
Guess a number: 25
Guess higher
Guess a number: 40
Guess higher
Guess a number: 45
Guess higher
Guess a number: 48
```

2) (30p) Using for loops write a Python program which finds the number of duplicate integers in a given array. Integers in the array will be between 0 and 99, both inclusive. Example arrays and the corresponding outputs are shows below. Notice that even if an integer appears more than twice, it is only counted as one duplicate.
Hint: You may use a second array to check uniqueness of a number.

Example outputs:

1) A=[73, 35, 67, 29, 56, 30, 83, 79, 82, 99, 19, 21, 78, 18, 19, 33, 9, 36, 81, 32, 43, 50, 34, 52, 9, 67, 67, 40, 89, 86, 7, 25, 30, 13, 25, 21, 96, 2, 91, 87, 45, 33, 9, 26, 90, 65, 61, 49, 94, 76]
   `There are 5 duplicate numbers`

2) A=[9, 38, 44, 12, 97, 7, 60, 69, 49, 40, 18, 94, 20, 94, 91, 70, 78, 11, 0, 41, 62, 93, 61, 40, 50, 59, 44, 57, 65, 16, 64, 39, 55, 27, 43, 69, 70, 94, 85, 43, 45, 57, 89, 8, 62, 3, 8, 23, 86, 94]

```
There are 4 duplicate numbers
```

3) (40p) Write a Python script that reads in n, m, and p and the matrix entries, then prints matrix A and B and product matrix AB.

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix}$$

the **matrix product** $\mathbf{AB}$ (denoted without multiplication signs or dots) is defined to be the $n \times p$ matrix[3][4][5][6]

$$\mathbf{AB} = \begin{pmatrix} (\mathbf{AB})_{11} & (\mathbf{AB})_{12} & \cdots & (\mathbf{AB})_{1p} \\ (\mathbf{AB})_{21} & (\mathbf{AB})_{22} & \cdots & (\mathbf{AB})_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{AB})_{n1} & (\mathbf{AB})_{n2} & \cdots & (\mathbf{AB})_{np} \end{pmatrix}$$

where each i, j entry is given by multiplying the entries A_{ik} (across row i of $\mathbf{A}$) by the entries B_{kj} (down column j of $\mathbf{B}$), for $k = 1, 2, ..., m$, and summing the results over k:

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj} .$$

Sample Test & Output:

```
sgoren@ubuntu:~/CSE101-Python/hw1$ ./hw1.py
2
3
4
Entries of matrix A :1 2 3 4 5 6
Matrix A
1 2 3
4 5 6
Entries of matrix B :2 2 2 2 3 3 3 3 4 4 4 4
Matrix B
2 2 2 2
3 3 3 3
4 4 4 4
Matrix Product
20 20 20 20
47 47 47 47
```