ENGRECE 12: Assignment-4

January 30, 2004

Due Wednesday 2/11/2004 12:00pm

1 Matrix Addition [20 points]

A matrix is a two-dimensional data structure. It is usually organized as rows of numbers. Examples of matrices include

$$A = \begin{pmatrix} 2 & 7 & 3 \\ 5 & 4 & -1 \\ 0 & 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 5 & 8 \\ 7 & 1 & 2 \\ 10 & 0 & -1 \end{pmatrix} \quad C = \begin{pmatrix} 2 & 5 \\ 7 & 4 \\ 3 & -1 \end{pmatrix}$$

A and B are 3-by-3 matrices, whereas C is a 3-by-2 matrix. In general, the rows and columns may be arbitrarily large. Each element of a matrix can be addressed by the row, column indices. Mathematically written as $a_{i,j}$, the subscript i is the row index, and j is the column index.

Two matrices can be added. Mathematically, $A + B$ is defined to be a new matrix S whose elements $s_{i,j} = a_{i,j} + b_{i,j}$. For instance, we have

$$\begin{pmatrix} 2 & 7 & 3 \\ 5 & 4 & -1 \\ 0 & 1 & 2 \end{pmatrix} + \begin{pmatrix} 3 & 5 & 8 \\ 7 & 1 & 2 \\ 10 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 5 & 12 & 11 \\ 12 & 5 & 1 \\ 10 & 1 & 1 \end{pmatrix}$$

In Python, a matrix can be represented as a list of lists (if mutable) or a tuple of tuples (if immutable). For instance, we can represent the matrices A and B above in Python as

```
A = [[2, 7, 3],
     [5, 4, -1],
     [0, 1, 2]]
B = [[3, 5, 8],
     [7, 1, 2],
     [10, 0, -1]]
```

Lists and tuples are data structures that can be used to store the values of the elements of the matrices. However, if we want to perform matrix operations such as addition, we cannot simply say `A + B`, because the **+** operator considers `A` and `B` to be lists, where **+** means joining these lists:

```
>>> A = [[2, 7, 3], [5, 4, -1], [0, 1, 2]]
>>> B = [[3, 5, 8], [7, 1, 2], [10, 0, -1]]
>>> A + B
[[2, 7, 3], [5, 4, -1],[0, 1, 2], [3, 5, 8], [7, 1, 2],[10, 0, -1]]
>>>
```

Which is NOT what we want as matrix sum. However, we can define a function that interprets `A` and `B` as two matrices and returns a new matrix that is their matrix sum.

Write a function `matrixAdd(A, B)` for computing the matrix sum. This function should contain the following:

- Create a blank new matrix

- Outer loop to iterate over each row i
- Inner loop to iterate over each column j of row i
- Add the corresponding $A[i][j]$ and $B[i][j]$, and store the sum in the new matrix
- After exiting the outerloop, return the new sum matrix

Your program should be saved in `matrix.py`. It should contain the `matrixAdd` function, followed by your test cases that define matrices, pass them to `matrixAdd`, and print the results.

2 Phone book [20 points]

This program helps you to understand the dictionary. A dictionary is essentially an advanced list that allows access to its elements by keys such as numbers or strings.So it can be used to keep some information like a phonebook. The objective of this program is to write a program in Python called `phonebook.py`. The program should have a menu with the following features:

- `Add  To add new name and number`
- `Delete  To delete a name and its number`
- `Change  To change a number of an existing name`
- `Lookup  To get a name and return its corresponding number`
- `Exit  To exit the program`

A sample session with the phone book program should look as follows:

```
Main menu
---------
1. Add new name and number
2. Delete a name and number
3. Change a the number of an existing name
4. Lookup the number associated with a name
5. Exit

Enter choice: 1
Entering a new phone book entry:
Name?   Jack
Number? 1234567

Main menu
---------
(as above)

Enter choice: 2

Deleting an entry:
Name?   Jack
Entry for "Jack" deleted.

Main menu (as above)

Enter choice: 2
Deleting an entry:
```

```
Name?   Jack
No such entry found.

Main menu (as above)

Enter choice: 3

Modifying an entry:
Name?   Jack
Old number is 1234567
New number? 3456789
Entry for Jack updated.

Main menu (as above)

Enter choice: 4

Looking up a number:
Name?   Jack
Jack's number is 3456789.

Main menu (as above)

Enter choice: 5

Exiting:
Are you sure to quit (Y/N)? Y
Exit.
```

Implement the phone book program in Python. Make sure that data entered by the user is properly checked and the program responds appropriately. For example, if a number is looked up for a user that does not exist, the program should state this fact and continue running (with the main menu) instead of aborting with an error message. Also, if a menu choice is entered outside the range of 1-5, the program should just repeat the main menu.

In order to test your program, run the above example session. Then, extend the session with the following tasks:

```
- add Jane with number 3213214
- add John with number 7778888
- lookup number for Jane
- lookup number for Joe (which does not exist!)
- modify number of Jane to 9990000
- lookup number of Jane
- delete number of Jack
- lookup number of Jack (should now not exist any more!)
- ...
```

3 What to turn in

Use the command

```
% python ~ece12/tools/submit.pyc
```

to turn in your homework. Your files should be a level above the `hw4` directory. You should have the following files:

- `matrix.py  matrix.script`
- `phonebook.py  phonebook.script`