Major Assignment 3

ENGINEER 1D04

Dr. William Farmer and Dr. Spencer Smith

McMaster University, Winter 2012

Revised: June 19, 2013

Please use AutoMarker (`automarker.mcmaster.ca`) and Avenue to acquire, test, package and submit your assignment. The procedure for submitting assignments is summarized on Avenue, with additional details provided by AutoMarker. **Please frequently back up your work by creating a submission package in AutoMarker**. This will provide a chance to recover your work in the event of an equipment failure.

Background

In this assignment you will implement a library of functions that can be used when processing a matrix, where a matrix $\mathbf{A}$ is a table of m rows and n columns as follows:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_{m-1} \end{bmatrix}$$

$$\text{where } \mathbf{r}_i = [a_{i,0} \quad a_{i,1} \quad \cdots \quad a_{i,n-1}]$$

An entry $a_{i,j}$ means the ith row and the jth column of the matrix $\mathbf{A}$. As an example:

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 6 & 4 \\ 1 & 2 & 3 & 3 \\ 9 & 1 & 6 & 7 \end{bmatrix} \text{ where } \mathbf{r}_0 = [3 \;\; 5 \;\; 6 \;\; 4], \mathbf{r}_1 = [1 \;\; 2 \;\; 3 \;\; 3], \text{and } \mathbf{r}_2 = [9 \;\; 1 \;\; 6 \;\; 7]$$

Design, implement, and test a program that satisfies the requirements below.

****IMPORTANT!!!**:** This assignment will be run through an automated testing program to be graded. Function syntax in your program must be **exactly** as specified, including spelling, capitalization, and the order of function parameters. **DO NOT** include any `input` or `print` statements, or a `main` function. Failure to precisely follow the requirements below will result in a **significant loss of marks**.

Requirements

1. Implement `sumPartial(x, d)` which takes a list `x` (of length `n`) and returns $\sum_{i=0,i\neq d}^{n-1} x_i$. That is, all entries in the list are summed together, except for the entry with the index d. For example, `sumPartial([5, 2, 1], 1) = 6`

2. Implement `extractCol(A, j)` which takes a matrix `A` and a column number `j` and returns a list of the entries in the `j`th column of `A`. Below is the general form of the output and then a specific example using the sample **A** given above and $j = 1$:

   in general for column j: $\begin{bmatrix} a_{0,j} \\ a_{1,j} \\ \vdots \\ a_{m-1,j} \end{bmatrix}$ specifically for $j = 1$: `[5, 2, 1]`

3. Using the previous two functions, implement `offDiagColSums(A)` which takes a matrix `A` and returns a list of the off diagonal column sums of `A`. That is, the following list is returned, first in general and then specifically for the example **A**:

   $$\left[\ \sum_{i=1}^{m-1} a_{i,0} \ \cdots \ \sum_{i=0,i\neq j}^{m-1} a_{i,j} \ \cdots \ \sum_{i=0,i\neq n-1}^{m-1} a_{i,n-1}\ \right]$$

   or for the example given `[10, 6, 9, 14]`.

4. The program requires very little besides the function definitions. There is no `main()`.

5. The program does not read anything from standard input or write anything to standard output. That is, the program does not interact with the user who invokes it.

6. The program is written in Python in a module, NOT in the Python Shell. To create a new module in IDLE, go to File → New Window. You must save this file with a .py extension. For more information on submitting your program, click the "AutoMarker Instructions" button above.

7. Your name, MacID, student number, and the date are given in comments at the top of your Python (.py) file before your program.

8. Your answers to the design questions and the test plan (see below) are given in comments at the bottom of your Python (.py) file after your program.

9. Your program MUST have valid Python syntax and it must run without errors. Ensure that your program runs properly by running it before you submit.

10. You must sign out with a TA or IAI after you have submitted your lab at the submission station. Failure to do so could result in a zero.

Design and Implementation Instructions

1. Conditional statements are not needed for this program.
2. The matrix is assumed to be implemented as a list of lists. Specifically it will be a list of rows, where each row is a list of values. That is, $\mathbf{A}$ is the list $[\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_{m-1}]$, where $\mathbf{r}_i$ is the list of values in the ith row of the matrix. For the example `A = [[3, 5, 6, 4], [1, 2, 3, 3], [9, 1, 6, 7]]`
3. You may assume that all rows are of the same length, that the matrix input consists of all numeric types, that inputs will be within a valid range and that error checking is not necessary.
4. Follow the function syntax **EXACTLY** as given, including spelling, capitalization and the order of function parameters.

Design Question

If `sumPartial` is implemented without a conditional statement, how many loops are required? What are the index ranges for these loops?

Test Plan

Produce a test plan with test cases for each of the functions in your library.

```
Test:  i for function j
Input:  inputs for function j
Expected Output:  expected output for function j
```

You should have enough test cases to adequately support the argument that your code is correct. Your test cases should cover as many different classes of input cases as possible, including boundary cases. Your test plan should include case(s) where your expected output is a failure.