CS11 Machine Problem

Matrices

Due: 25 August 2021

1 Introduction

Matrices can be treated as a rectangular array of numbers or expressions arranged in rows and columns. They are widely used in mathematics and in other sciences and engineering fields. In this activity, you will create a module that has functions containing the basic operations of matrices.

2 Specification

Create the following modules. You have the liberty to design your function according to the requirements/specifications.

a) $matrix.py$: the module that contains operations on matrices
The dimension of a matrix A is defined by the number of rows m and the number of columns n. A can be also written as $A_{m\times n}$. The following is an example of a 2×3 matrix.

$$\begin{bmatrix} 5 & -3 & 2 \\ 4 & 1 & 8 \end{bmatrix}$$

An element a_{ij} of a matrix denotes the element in the ith row and jth column. Example: $a_{12} = 4$.

The $matrix$ module should contain the following basic operations on matrices

i) add
The sum of two matrices $A_{m\times n}$ and $B_{m\times n}$ with the same dimension is an $m \times n$ matrix whose elements are the sum of the respective elements of A and B.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \end{bmatrix}$$

ii) scalar_multiply
Given a matrix $A_{m\times n}$ and any number k, then the scalar product of k and $A_{m\times n}$ is an $m \times n$ matrix whose elements are obtained by multiplying k to each element of A.

$$k\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} ka_{11} & ka_{12} & ka_{13} \\ ka_{21} & ka_{22} & ka_{23} \end{bmatrix}$$

iii) multiply
Given matrices $A_{m\times p}$ and $B_{p\times n}$ where the number of columns of A is equal to the number of rows of B, their product C=[c_{ij}] where element

$$c_{ij} = \begin{bmatrix} a_{i1}\ a_{i2} \dots a_{ip} \end{bmatrix} \cdot \begin{bmatrix} b_{1j} \\ b_{2j} \\ \dots \\ b_{pj} \end{bmatrix} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ip}b_{pj}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

iv) transpose

Given a matrix $A_{m \times n}$, the transpose of A (denoted as A^T) is an $n \times m$ matrix whose elements are flipped over its diagonal, that is, each element a_{ij} in A is the element a_{ji} in A^T

The transpose of the matrix $\begin{bmatrix} 5 & -3 & 2 \\ 4 & 1 & 8 \end{bmatrix}$ is the matrix $\begin{bmatrix} 5 & 4 \\ -3 & 1 \\ 2 & 8 \end{bmatrix}$

b) $main.py$: the main program which handles the reading input, calling the functions of matrix to process the input and then writing the output

a. Input

Input will be read from a file. The input has the following format

Line 1 : operation name (add,scalmultiply,multiply,transpose)

Line 2: dimension of matrix A

Line 3-x: elements of matrix A

Line x: dimension of matrix B (the scalar number for scalmultiply; not needed for transpose)

Line x+1 - y: elements of matrix B (not needed for transpose and scalmultiply)

b. Output

Output will be written also in a file.

Sample Input 1:

```
add
2 2
-88 73
-55 54
2 2
-10 54
-18 -58
```

Sample Output 1:

```
-98 127
-73 -4
```

Sample Input 2:

```
scalMultiply
2 2
-10 5
4 8
3
```

Sample Output 2:

```
-30 15
12 24
```

Sample Input 3:

```
multiply
3 3
2 1 3
3 -2 1
-1 0 1
3 2
1 -2
2 1
4 -2
```

Sample Output 3:

```
16 -12
3 -10
3 0
```

Sample Input 4:

```
transpose
2 3
2 1 3
3 -2 1
```

Sample Output 3:

```
2 3
1 -2
3 1
```

3 Deliverables

1. In your python scripts, at the top, annotate the names of your group members, along with their student number and section.

2. Create a very short video(3-5 mins) that shows a walkthrough of your project.

3. Criteria for grading is based on, but not limited to,

a) Correctness (40%)
   - perform the matrix operations correctly for any valid input

b) Functionality (30%)
   - use functions and modules correctly
   - design functions and modules such as they can be reusable

c) Error Handling (20%)
   - handle invalid input
   - display and handle possible error

d) Coding Style/Readability (10%)
   - annotate important parts
   - use of descriptive names