Middle East Technical University

Department of Computer Engineering

CENG 111

Fall 2015

Take Home Exam 3

v1.2

REGULATIONS

Due date: 23:59, 06 January 2015, Wednesday *(Not subject to postpone)*

Submission: Electronically. You should save your program source code as a text file named `the3.py` and submit it to us via the course's COW page.

Team: There is **no** teaming up. This is an EXAM.

Cheating: Source(s) and Receiver(s) will receive zero and be subject to disciplinary action.

PROBLEM

This homework is about implementing matrix algebra in the rational-number domain. You are expected to write a function named `mateval`, which will take an expression in the prefix notation and return the result of the evaluation. In this prefix expression, there will be matrices and rational numbers operated by `+` (addition), `-` (subtraction) and `*` (multiplication) operations. Moreover, the prefix expression might involve multiplication of matrices by rationals or integers. However, addition/subtraction of rationals or integers to/from matrices is not allowed. We require that the rules for matrix algebra have to hold (like the requirement of dimensional equality in addition/subtraction, etc.).

To be more precise:

- The representation of rational numbers will be as a list of two integers: The integer value of the numerator, followed by a **positive** integer value of the denominator (different than 1). Our rational numbers do not include integers. Moreover, if a rational number has any common multiplier between the numerator and the denominator, it is removed. In other words, the greatest common divisor of the numerator and the denominator should always be 1 and kept as 1 if it is not. Examples:

  `[4,13]` represents $\frac{4}{13}$

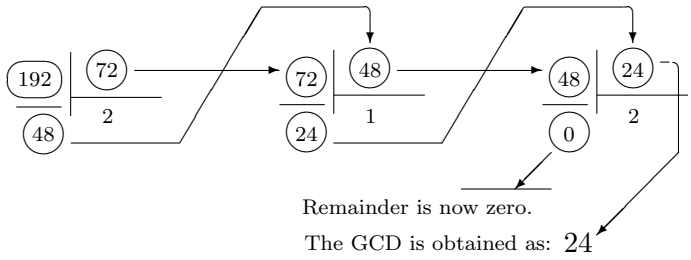
  `[-1,13]` represents $-\frac{1}{13}$

  `[1,-2]` is not allowed.

  `[6,1]` is not allowed.

  `[12,36]` is not allowed.

You should use the following algorithm for finding the GCD *(greatest common divisor)*. Let us assume we want to find GCD(192,72):



- The representation for matrices is a list of lists (each of which represents a row).
  The element of a matrix can be either an integer or a rational. A 1×1 matrix is invalid.

  Example:

Representation	**Corresponding matrix**
`[[2,-3],[1,5]]`	$\begin{bmatrix} 2 & -3 \\ 1 & 5 \end{bmatrix}$
`[[2,[-1,2]],[[7,3],5]]`	$\begin{bmatrix} 2 & -\frac{1}{2} \\ \frac{7}{3} & 5 \end{bmatrix}$
`[[-8,[-1,2],[3,25]],[[7,3],[1,5],[-9,23]]]`	$\begin{bmatrix} -8 & -\frac{1}{2} & \frac{3}{25} \\ \frac{7}{3} & \frac{1}{5} & -\frac{9}{23} \end{bmatrix}$
`[[2,3]]`	$\begin{bmatrix} 2 & 3 \end{bmatrix}$
`[[2],[3]]`	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
`[[[2,3],1]]`	$\begin{bmatrix} \frac{2}{3} & 1 \end{bmatrix}$

SPECIFICATION

- You should name your main function that does the evaluation as `mateval`. You may define additional functions, of course.
- You should implement the `+`, `-`, `*` operations among rationals and/or integers.
- You should also implement matrix operations of `+`, `-`, `*`. Among these, `+` and `*` can be n-ary except unary (unary usage of `+`, `*` is not allowed). Subtraction (`-`) is strictly binary. In addition to the operations among matrices, multiplication of a matrix by a rational or an integer is also allowed.
- As you see in the example given in the next section, each operation is *contained* in a list such that the first item of the list is the operator, and the other items are the operands. In other words, we have a nested structure.
- There is no restriction on the size of matrices except that 1×1 matrices are invalid.
- You should try to write short and concise functions.
- The rules that define a rational (given in the previous section) have to hold for the value returned by `mateval`. In other words, any rational returned as a value or a sub-part of the value of a `mateval` call, has to comply fully with the definition of a rational.
- If the result of `mateval` is a rational which is equal to an integer then you are supposed to return the integer.
- The input, which your functions will be tested with, will be error-free. You do not have to perform any error check. Moreover, no 1×1 matrix will be a result.
- Do not use floating points ever!
- You are not allowed to use any Python module for any reason.
- Your function(s) should not print anything on the screen.
- You may not use any data type or mechanism of Python not covered in the lectures or the labs. In other words, you may not use dictionaries, or classes.

EXAMPLE RUN

$$\left(1-\frac{1}{3}\right)\times\begin{bmatrix}2 & -\frac{1}{2}\\ \frac{7}{3} & 5\end{bmatrix}\times\begin{bmatrix}-8 & -\frac{1}{2} & \frac{3}{25}\\ \frac{7}{3} & \frac{1}{5} & -\frac{9}{23}\end{bmatrix}+\begin{bmatrix}2 & 3 & -1\\ \frac{1}{2} & \frac{1}{3} & -1\end{bmatrix}=\begin{bmatrix}-\frac{85}{9} & \frac{34}{15} & -\frac{408}{575}\\ -\frac{25}{6} & \frac{2}{9} & -\frac{3653}{1725}\end{bmatrix}$$

```
>>> mateval(['+',['*',['-',1,[1,3]],
                      [[2,[-1,2]],[[7,3],5]],
                      [[-8,[-1,2],[3,25]],[[7,3],[1,5],[-9,23]]]],
                  [[2,3,-1],[[1,2],[1,3],-1]]])

[[[-85,9],[34,15],[-408,575]],[[-25,6],[2,9],[-3653,1725]]]
```