

We live in a world of massive data generation, but what happens to all of that data? How do we analyze it? Enjoying Netflix?, Amazon?, Pandora? The computational backbone to all of these? Matrix algebra and some highly specialized algorithms. Where do you fit in? You and your team<sup>1</sup> have been charged with creating a new matrix class (uMatrix) in C++ that will end up on an ultra-portable device. As such, your team only has access to one library: `iostream`. You may also increase the readability of your code with `using namespace std;`.

**The problem:** You must grab data directly piped in from the command line <sup>2</sup>. The data coming from the pipe is guaranteed to form a square matrix, and is organized in comma-separated, row-major format. e.g. the matrix:

$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$$

Would be passed in as A,B,C,D,E,F,G,H,I [ret]

Since the length of this input is 9, it forms a 3 by 3 matrix.

Your class must support any integer matrix up to 100 x 100<sup>3</sup>

Data will always be provided in the following format:

<MATRIX A> [ret]

<MATRIX B> [ret]

<SCALAR S> [ret]

1,2,3,4,5,6,7,8,9

9,2,3,4,5,1,2,3,4

2

You must implement the following functionality (twice):

1. matrix multiplication (75% likely)
2. scalar multiplication (10% likely)
3. matrix addition (10% likely)
4. scalar addition (5% likely)

For each of these functions you should build two versions, one that is "straight-forward/slow" and another that leverages some matrix iteration short-cut <sup>4</sup>.

#### DEADLINE:

Your customer, Dr. Borowczak, is asking for this uMatrix class **no later than March 18th @ 11:59PM Mountain Time**. There is a 5% bonus for submitting the class to your customer, Dr. Borowczak, via WyoWeb by March 11th @ 11:59PM Mountain Time.

<sup>1</sup>max team size = 2

<sup>2</sup><https://stackoverflow.com/questions/5446161/reading-piped-input-with-c>

<sup>3</sup>The customer states that if you can support matrices of 100,000 x 100,000 they will give you a 5% bonus!

<sup>4</sup>[http://nadeausoftware.com/articles/2012/06/c\\_c\\_tip\\_how\\_loop\\_through\\_multi\\_dimensional\\_arrays\\_quickly](http://nadeausoftware.com/articles/2012/06/c_c_tip_how_loop_through_multi_dimensional_arrays_quickly)

## 1. Requirements

- ☐ Delivery of a well commented uMatrix class (.h/.cpp) (10pt)
- ☐ Driver main (yourName.cpp) (10pt)
  - ☐ Reads in 2 matrices (A,B) and a scalar (s) into 2 uMatrix object
  - ☐ Reads A,B, and S from the command line (non-interactive)
- ☐ constructor, destructor, print (15 pt)
- ☐ matrix multiplication (30pt)
  - ☐ slow (10 pt)
  - ☐ fast (20 pt)
- ☐ scalar multiplication (5pt)
  - ☐ slow (1 pt)
  - ☐ fast (4 pt)
- ☐ matrix addition (10pt)
  - ☐ slow (3 pt)
  - ☐ fast (7 pt)
- ☐ scalar addition (5pt)
  - ☐ slow (1 pt)
  - ☐ fast (4 pt)
- ☐ Discretionary & Analysis (15pt)
  - ☐ Project Readme.md in github with analysis of each function (n=11) (11pt)
  - ☐ Customer's Choice (4pt)