

Engineering Programming 1

Project 2: Matrix Calculator

Fall 2015

Due date: Dec 18 (23:59 pm)

(revised: Dec 7)

Project Description

- You should implement a simple matrix calculator.
- The *maximum* score you can get from this project is **200** points, if you implement all the requirements.
- **Do everything by your own exertion!**

Project Description

- **You should implement the following functions:**
 - matrix creation
 - matrix deletion
 - matrix transpose
 - matrix addition
 - matrix multiplication
 - matrix scalar multiplication
 - to save/load/show matrices

Matrix Creation

- When you create a matrix, make sure that your matrix variable is created in the *heap space*. That is, you should use “**malloc**” or “**new**” to create a matrix. All the matrices in this project should be

row-major order.

- You should implement 3 different creation functions:

zeros

create and fill all the elements in the matrix with 0.

ones

create and fill all the elements in the matrix with 1.

rand

create and fill all the elements in the matrix with randomly generated numbers.

- **Example**

A = zeros(3, 5); // A is a matrix with 3 rows and 5 columns

B = rand(4, 4); // B is a matrix with 4 rows and 4 columns

Matrix Deletion

- Since your matrix variable is in the *heap space*, you should be in charge of deleting it when the variable is no more needed. Make sure that the memory space for the input matrix to be deleted is completely deallocated.
- **delete**
delete the given matrix variable.
- ***Example***
delete(A)

Matrix Transpose

- The output of the transpose function should be a newly allocated matrix, i.e., you should create a new matrix inside the transpose function and return the matrix whose elements are filled with those of transpose of the matrix A.
- You should make sure that the other functions like addition and multiplication to be implemented in this way.
- **transpose**
transpose given matrix variable.
- **Example**
 $R = \text{transpose}(A)$

Matrix Addition

- Matrix addition function will take two or three inputs(bonus points) and add them together. Make sure that the size of the input matrices are equal. Otherwise the return value should be **NULL**.
- **add**
add all the given matrices.
- **Example**
 $R = \text{add}(A, B)$
 $R = \text{add}(A, B, C)$

Matrix Multiplication

- Matrix multiplication function will take two or three inputs(bonus points) and multiply them together. Make sure that the size of the input matrices are valid to be computed. Otherwise the return value should be **NULL**.
- **multiply**
multiply all the given matrices.
- **Example**
 $R = \text{multiply}(A, B)$
 $R = \text{multiply}(A, B, C)$

Matrix Scalar Multiplication

- Matrix scalar multiplication function will take a matrix variable and a scalar variable, and multiply them.
- **multiply** (use function overloading)
multiply all the given matrices.
- ***Example***
 $R = \text{multiply}(A, c)$ // A is a matrix and c is a scalar

Matrix Load/Save

- As the name suggests, matrix save/load function will save/load the computed a matrix into/from a formatted file. There's no specific format for these functions. You may define the format by yourself.

- **load**

load a matrix with the given filename.

- **save**

save a matrix into a file with the given filename.

- ***Example***

`load("A.txt")` // *load the matrix A from the file "A.txt"*

`save(A, "A.txt")` // *save the matrix A into the file "A.txt"*

Displaying a Matrix

- You should implement a function to show a matrix.
- **show**
show the given matrix. There should be 3 spaces between each element.
- **Example**
show(A) // show the matrix A

1 2 3 // result of show function with
2 3 4 // 2 by 3 matrix A (Row-major matrix)

Basic Requirements (100 points)

Creation	Deletion	Transpose	Addition with two inputs	Multiplication with two inputs
20 pts	10 pts	10 pts	10 pts	15 pts

Scalar Multiplication	Save	Load	Show
5 pts	10 pts	10 pts	10 pts

Bonus Points

(100 points)

Binary file save/load (10 pts)	Implement <i>bsave/bload</i> for binary file to be saved/loaded. (You should use <i>ios::binary</i> option)
Addition/Multiplication with three inputs (20 pts)	Implement addition/multiplication with three inputs.
Efficient multiplication with three inputs (20 pts)	Considering the order of multiplication of given three matrices, you should implement the multiplication with three inputs efficiently. For example, A is a 100000 x 5 matrix, B is a 5 x 5 matrix, and C is a 5x1 matrix. In this case, $(A*(B*C))$ will be much more efficient than $((A*B)*C)$
Sparse Matrix (50 pts)	Implement all the functions specified with sparse matrix form. If you don't know what sparse matrix is, you can refer to the Wikipedia (https://en.wikipedia.org/wiki/Sparse_matrix)

Submission

- The skeleton code will be given. It consists of three files: main.cpp, matrix.h, and matrix.cpp
- **main.cpp**
You can test the functions which you implemented in the main function. The sample test code is given.
- **matrix.h**
Do not touch this file!!
- **matrix.cpp**
You should implement the functions in this file.
- **Submit **matrix.cpp** only!**

How to Submit

- First, log in to your account.
- Second, use the following command to submit the file.
- Note that the system will be automatically closed at the deadline!

submit proj2 matrix.cpp

- Submission method pattern is same with Project1. (**Do not use -t option!**)

How To Compile

- `g++ main.cpp matrix.cpp -o $(exec_name)`
- `./$(exec_name)`
- replace ***\$(exec_name)*** with your execution file name.

No plagiarism!

- According to our academic dishonesty policy, all projects must be done entirely by yourself, without the help of any other person.
- We will enforce this policy by using a professional-grade program checker to catch students who commit plagiarism. **Students who get caught will receive zero points in the project and/or fail this course immediately.** We will also inform the university about the incident.

No plagiarism!

- You must protect your programs from being seen or copied by other people. In case if your program is copied by other students without your acknowledgment, **you will still be considered committing plagiarism.**
- You cannot copy any program that you found on the Internet. Even copying a small fragment of code is not allowed.
- **If you notice any plagiarism activity, please report to the instructors or TAs.**