

COSC2430 Hw0: User-Defined Dense Matrix Addition

1. Introduction

You will create a C++ program to add two matrices. You can program matrix addition with any algorithm, provided results are correct. It is preferred your algorithm can add matrices efficiently, avoiding unnecessary products or data movement in the memory hierarchy.

Given two matrix A and B of size $m \times n$ you will compute result matrix $C = A+B$.

2. Input and Output

The input is two text files, each with one matrix. The output is a single text file, with one matrix.

File format: There will be ONE matrix row per line in the file. Each value is a real number that may or may not have a decimal point (e.g. 1, 2.1, 3.1416). Values are separated by spaces. Except for digitals, no other character will be given.

3. Program and input specification

The main C++ program will become the executable to be tested by the TAs. The result matrix should be left on another text file (output file), provided on the command line. Notice the input and output files are specified in the command line, not inside the C++ code. Notice also the quotes in the program call, to avoid

Unix/Windows get confused.

The general call to the executable is as follows:

```
denseadd "A=<file>;B=<file>;C=<file>"
```

Call example with one input file and another output file.

```
denseadd "A=a.txt;B=b.txt;C=c.txt"
```

Assumptions: Matrices are given in dense form. Real numbers are separated by one space. The number of column and row in the smaller column and row determines matrix size $m \times n$. The output matrix will be written in dense form with a fixed number of decimals.

Example 1 of input and result matrix

1.txt (A=1.txt)

```
1 0
0 1
```

2.txt (B=2.txt)

```
1 0
0 1
```

3.txt (C=3.txt output file)

```
2.00 0.00
0.00 2.00
```

Example 2 of input and result matrix

1.txt (A=1.txt)

```
0.5 3.0 0.0 0.0
0.0 1.0 0.8 0.8
1.0 0.0 0.2 0.2
```

2.txt (B=2.txt)

```
1.0 6.0 0.0
0.0 2.0 1.6
2.0 0.0 0.4
```

3.txt (C=3.txt output file)

```
1.50 9.00 0.00
```

```
0.00 3.00 2.40
3.00 0.00 0.60
```

Example 3 of input and result matrix

1.txt (A=1.txt)

```
0.5 3.0 0.0 0.0
0.0 1.0 0.8 0.8
```

2.txt (B=2.txt)

```
1.0 6.0
0.0 2.0
2.0 0.0
```

3.txt (C=3.txt output file)

```
1.50 9.00
0.00 3.00
```

4. Requirements

- C++:

It is encouraged you do not use existing STL, vector classes since you will have to develop your own C++ classes and functions in future homework.

Your C++ code must be clear, indented and commented.

You must determine matrix size based on number of columns of the first row of the input matrix.

You can use static arrays of a maximum size=20. Your program will be tested with matrices up to 20 x 20. You can optionally use dynamically-sized 1-dimensional arrays (not dynamic 2D arrays since those require more careful manipulation) whose size is determined at run time.

Include comments on each source code file and a README file on how to compile/run with examples, as obvious it may seem.

Your program will be tested with GNU C++. Therefore, you are encouraged to work only on Unix. You can use other C++ compilers, but the TAs cannot provide support or test your programs with other compilers.

- Output:

The output file must contain the result, in the same format (one row per line).

Matrix values are written with 2 decimals separated by one space. Do not use other separators or different number of decimals. Notice the number of decimals may vary in future homework.

Your program should write error messages to the standard output (cout, printf).