

# COSC2430: Programming and Data Structures

## Matrices: sum of rows and columns

### Introduction

You will create a C++ program to compute the sum of rows and columns of a square matrix. You are encouraged to develop an algorithm that computes the sums using only one nested loop.

### Input and Output

The input is a single text file, with one matrix. The output is also a single text file, with two rows (first row is the sum of rows, second row is the sum of columns).

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. Comments have a # at the beginning of the line (no spaces before).

### Example 1 of input and result

#Matrix A, size 2

1 5

3 1

6.00 4.00

4.00 6.00

### Example 2 of input and result

# Matrix A, size 3

0.5 3.0 0.0

0.0 1.0 0.8

1.0 0.0 0.2

3.50 1.80 1.20

1.50 4.00 1.00

### Program and input specification

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

The general call to the executable (sum\_rowcol, in this example) is as follows:

```
sum_rowcol "A=<file>;C=<file>"
```

Call example with one input file and another output file.

```
sum_rowcol "A=a.txt;C=c.out"
```

### Requirements

- You are encouraged to develop an algorithm that computes the sums using only one nested loop. Notice that the input matrix A is assumed to be squared.
- For this homework, it is allowed to use vector classes. In the future, use of STL will be forbidden since you will have to develop your own C++ classes and functions.
- Your C++ code must be clear, indented and commented.
- You can determine matrix size based on number of columns of the first row of the input matrix, or you can use the information in the first row of the file (commented, starts with #). You can assume that the first line (commented) will include only one number, but no assumptions should be made on its position (it will not necessarily be at the end of the line). Your program must reject input matrices with errors producing an empty file.
- You can use static arrays of a maximum size=20. Your program will be tested with matrices up to  $20 \times 20$ .
- Your program will be tested with GNU C++. Therefore, you are encouraged to work on Unix, or at least make sure that your code compiles and runs successfully on the server.
- You can use other C++ compilers, but the TAs cannot provide support or test your programs with other compilers.
- The output file must contain the result, in the same format (one row per line). Results are written with 2 decimals separated by one space. Do not use other separators or a different number of decimals. Notice the number of decimals may vary in future homework.
- In case of invalid input or other exceptions, your program should output an empty file, and write an appropriate error message to the standard output (cout, printf).

### Testing for grading:

- Your main cpp program will be automatically compiled. If there is an error in compilation it will not be executed. **Programs that do not compile on the server will receive a score of 0.**
- Your program should not crash, halt unexpectedly, take an unusual amount of time to run (more than 10 seconds) or produce unhandled exceptions. Test your program using empty input, zeroes and inconsistent information. Each

- Your program will be tested with 10 test cases, going from easy to difficult. You can assume 70% of test cases will be clean, valid input matrices.

## **DEADLINE: September 3<sup>rd</sup> (midnight)**

In general the due date cannot be changed. No exceptions, unless there are medical or exceptional reasons.

### **Grading**

- A program not submitted by the deadline is zero points.
- A program that compiles but does not work program is worth 10 points.
- A program that does some computations correctly, but fails several test cases (especially the easy ones) is worth 50 points. Only programs that work correctly with most input files that produce correct results will get a score of 80 or higher. In general, correctness is more important than speed.
- A code with no comments will receive a -5 points penalty.

Plagiarism and cheating: C++ code is individual: it cannot be shared (other than small fragments used in class or in lab examples). Programs will be compared for plagiarism. If the TAs detect that a portion of your C++ code is highly similar to C++ on the Internet or other student the grade will be decreased at least 50%.