Data Structures
BITL1

Worksheet 4

Fall 2020

1 October 2020

---

1. Recall the structure `Matrix` introduced last week.

   (a) Download the files `matrix.cpp` and `header.h`, and compile `matrix.cpp`.

   (b) Add code to the `main` function so that the matrices `mat1` and `mat2` are printed to the console in the expected manner, that is, in `m` rows and spaces separating each of the `n` columns.

2. This question will deal with **inheritance**.

   (a) Write a structure `SqSubMatrix` derived from `Matrix`, to indicate chosen square submatrix of some given matrix, that has two new public objects:

   - `size`, an integer between 1 and the minimum of `m,n`
   - `topleft`, an integer array of size 2

   (b) Give this structure a function `getSub`, which returns the `Matrix` structure that is the square submatrix of size `size` starting from the row-column position given by `topleft`.

   (c) Give this structure a function `getTrace`, which returns the trace of the square submatrix. Recall that the *trace* of a square matrix is the sum of its diagonal entries.

3. This question will deal with **polymorphism**. Create a new function `addSqSubMatrices` that takes in two and returns one `SqSubMatrix` structure that is the matrix sum of the two inputs. Also, it should:

   - call (nontrivially) the function `addMatrices`
   - give `size` to the structure it returns that is the smallest of the two `size`s of the inputs
   - give `topleft` to the structure it returns that is the smallest of the two `topleft`s of the inputs, where size measured first by first coordinate, and if the first coordinate is the same, then by the second coordinate.

4. This question will deal with **vectors**. The `Matrix` structure defined here can be thought of as a `vector` class with two integers whose product is its `size`.

   (a) Rewrite `multiplyMatrices` into a function `multVecAsMat` that takes in:

   - two `vector` class objects that represent the matrices to be multiplied, and
   - one integer, which represents the number of columns of the first and the number of rows of the second,

   and outputs one `vector` class object that is the multiplied "matrix".

   (b) Assume you are only multiplying square matrices. What is the complexity of your function, in terms of the number of rows (which is the same as the number of columns)?