San José State University
Department of Computer Engineering

CMPE 180A

Data Structures and Algorithms in C++

Spring 2018

Instructor: Ron Mak

Assignment #3

Assigned:	Friday, February 9
Due:	Thursday, February 15 at 5:30 PM
URL:	http://codecheck.it/codecheck/files/18021006071yha0s7ifn2odnn9za9j206hz
Canvas:	Assignment 3. Hilbert Matrices
Points:	100

Hilbert matrices

This assignment will give you practice working with one- and two-dimensional arrays.

Matrices

A *matrix* is a two-dimensional array. This assignment works with square matrices, where the number of rows equals the number of columns. This number is the *size* of the matrix.

An example of a matrix A of size 3: $A = \begin{matrix} 3 & 0 & 5 \\ -1 & 4 & 4 \\ 1 & -3 & 2 \end{matrix}$

An *identity matrix* I is a square matrix with 1's along the main diagonal and 0's elsewhere.

The size 3 identity matrix: $I = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$

The identity matrix has the property that multiplying any square matrix by the identity matrix of the same size yields the same square matrix: $A \times I = A$

(If you don't remember how to multiply matrices, see the tutorial at https://www.basic-mathematics.com/multiply-matrices.html.)

Matrix inverse

Given a square matrix A, the inverse of A, written A^{-1}, is the matrix such that $A \times A^{-1} = I$

The inverse of the example matrix A above is (approximately):

$$A^{-1} \approx \begin{matrix} 0.363636 & -0.272727 & -0.363636 \\ 0.109091 & 0.018182 & -0.309091 \\ -0.018182 & 0.163636 & 0.218182 \end{matrix}$$

Not every square matrix has an inverse. But if a matrix does have an inverse, we can use the *LU Decomposition Algorithm* to compute it.

(If you really want to know, see https://en.wikipedia.org/wiki/LU_decomposition.)

The Hilbert matrix

In 1900, the prominent German mathematician David Hilbert (1862-1943) presented his famous collection of 23 problems, all unsolved at the time, that set the course of much of mathematical research in the 20th century. He introduced the Hilbert matrix in 1894.

A Hilbert matrix H is a square matrix whose elements are not difficult to compute. Number its rows and columns 1 though n, where n is the size of the matrix, then

$$H_{i,j} = \frac{1}{i+j-1}$$

For example, a Hilbert matrix of size 3 is

$$H = \begin{matrix} 1/1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{matrix} = \begin{matrix} 1.000 & 0.500 & 0.333 \\ 0.500 & 0.333 & 0.250 \\ 0.333 & 0.250 & 0.200 \end{matrix}$$

A feature of a Hilbert matrix is that it is especially challenging to compute its inverse accurately. For the Hilbert matrix above,

$$H^{-1} = \begin{matrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{matrix}$$

Elements of H^{-1} are all integers. The larger the Hilbert matrix, the greater the magnitude of the elements of its inverse.

What your program should do

In this assignment, your program should repeatedly loop. In each loop:

1. Prompt the user for the size of the Hilbert matrix, from 1 though 5.
2. Compute the elements of the Hilbert matrix H of the chosen size.
3. Use the LU decomposition algorithm to compute (an approximation of) the inverse H^{-1}. The C++ code for the algorithm will be provided for you.
4. Multiply $H \times H^{-1}$ to compute (an approximation of) the identity matrix I.
5. Invert H^{-1} to see how closely you get back H.

Terminate the loop and the program if the user enters a size outside of the range 1 - 5.

You will see all of the C++ code for the LU decomposition algorithm in CodeCheck. You must figure out how to call that code to compute a matrix inverse.

CodeCheck will supply integer values as input to your program. Therefore, your program should read a single integer value after each prompt.

Sample expected output

```
Size of Hilbert matrix (1-5)? 2

Hilbert matrix of size 2:

       1.000000       0.500000
       0.500000       0.333333

Hilbert matrix inverted:

       4.000000      -6.000000
      -6.000000      12.000000

Hilbert matrix multiplied by its inverse:

       1.000000       0.000000
       0.000000       1.000000

Inverse Hilbert matrix inverted:

       1.000000       0.500000
       0.500000       0.333333

Size of Hilbert matrix (1-5)? 5

Hilbert matrix of size 5:

       1.000000       0.500000       0.333333       0.250000       0.200000
       0.500000       0.333333       0.250000       0.200000       0.166667
       0.333333       0.250000       0.200000       0.166667       0.142857
       0.250000       0.200000       0.166667       0.142857       0.125000
       0.200000       0.166667       0.142857       0.125000       0.111111

Hilbert matrix inverted:

      25.000000    -300.000000    1050.000000   -1400.000000     630.000000
    -300.000000    4800.000000  -18900.000000   26880.000000  -12600.000000
    1050.000000  -18900.000000   79380.000000 -117600.000000   56700.000000
   -1400.000000   26880.000000 -117600.000000  179200.000000  -88200.000000
     630.000000  -12600.000000   56700.000000  -88200.000000   44100.000000

Hilbert matrix multiplied by its inverse:

       1.000000       0.000000       0.000000       0.000000       0.000000
       0.000000       1.000000       0.000000       0.000000      -0.000000
       0.000000      -0.000000       1.000000       0.000000       0.000000
       0.000000       0.000000       0.000000       1.000000      -0.000000
       0.000000      -0.000000       0.000000       0.000000       1.000000

Inverse Hilbert matrix inverted:

       1.000000       0.500000       0.333333       0.250000       0.200000
       0.500000       0.333333       0.250000       0.200000       0.166667
       0.333333       0.250000       0.200000       0.166667       0.142857
       0.250000       0.200000       0.166667       0.142857       0.125000
       0.200000       0.166667       0.142857       0.125000       0.111111

Size of Hilbert matrix (1-5)? 0

Done!
```

Note: The input values that CodeCheck supplies your program will not appear after the prompts in your program's output.

Submission into Canvas

When you're satisfied with your program in CodeCheck, click the "Download" link at the very bottom of the Report screen to download a signed zip file of your solution. Submit this zip file into Canvas. You can submit as many times as you want until the deadline, and the number of submissions will not affect your score. Only your last submission will be graded.

Submit the signed zip file from CodeCheck into Canvas:
Assignment #3. Hilbert Matrices.

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Program output (as verified by CodeCheck) • Correct output values. • Correct output format.	**35** • 25 • 10
Correct program design • Good functional decomposition. • Correct calls to the matrix inversion code. • Correct array code.	**50** • 15 • 15 • 20
Good program style • Descriptive variable names. • Meaningful comments. • Follows the coding style of the Savitch textbook (formatting, braces, indentation, function declarations before the main, etc.)	**15** • 5 • 5 • 5

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.