**CS 201R**

**Problem Solving & Programming II**

**Program 2 – Matrix Functions Due \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Key Learning Objectives:**

* Arrays & Vectors (2-dimensional)
* Functions (pass by value vs pass by reference)
* Looping
* Branching
* Print formatting
* Bonus: file input

**Assignment Problem:**

You will be creating a series of functions to complete this project. Below is the list of functions that you should create for full credit:

1. Create a function to initialize the matrix (sample matrix data is listed below)

d = 4;

m1 = { {1,1,1,0},

{1,1,0,1},

{1,0,0,0},

{0,1,0,1} };

d = 3;

m1 = { {1,1,0},

{0,1,0},

{1,0,1}};

d = 2;

m1 = {{1,0},

{0,1}};

1. Allow the user to pick which of the 3 dimensions they prefer and use that matrix to see if it is reflexive, symmetric, transitive, and to find the transitive closure. To do this, create the functions:
   * Create a function to print the matrix given
   * Create a function to add two matrices together (you will be using the same matrix 2 times)
   * Create a function to square the matrix given
   * Create a function to check if the matrix is reflexive (if all values on the diagonal are 1)
   * Create a function to check if the matrix is symmetric (values are equal across the diagonal)
   * Create a function to check if the matrix is transitive (not transitive if there is a non-zero in the squared matrix that is zero in the original matrix)
   * Create a function to find the transitive closure of the matrix (continue to multiply the matrix by itself up to the number of dimensions and keep a running total of the sum of all of these matrices)
2. An explanation of reflexive, symmetric & transitive can be found here: [Matrix Relations.ppt](https://umkc.box.com/s/1im8vs1bo3bi5vvmfl6c9qr81jq2weml)

& [Matrix Relations video](https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8fd9bce7-28a1-4559-8be6-afa901079e71). Transitive closure is described here: [Matrix Relations – Transitive Closure Video](https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=b6dcb55a-09d6-4f32-bde1-afa901105417)

**Bonus:**

1. Create a function read an input file (see [data.txt](https://umkc.box.com/s/l8r05hjzrq83qbr83j0s4inlz65b7al4) for input example). Add the code to read the matrix size & load the elements of a matrix (in row-major order) from a file, reading & processing multiple matrices and outputting the results
2. Prompt to see if user would like to verify a path. If so, allow user to enter a starting value and ending value to determine if a path exists from the starting point to the ending point (if the coordinate point in the transitive closer is non-zero, there is a path from i to j (i,j).

**Practical solution:**

1. Be sure to comment code
2. You must use functions (.h and .cpp files) for the functions listed above

**Submission:**

* Complete [THIS FORM](https://forms.gle/jkK7GBRxYB4XpRK4A) for turn in. You will need to include:
  + Your pseudocode. This can be a link to a google document, an image, or a miro page. Make sure to share this with [gladbachj@umkc.edu](mailto:gladbachj@umkc.edu) and <stgdcg@umsystem.edu>.
  + You will complete your code in Visual Studio. Please then copy your code to your repl.it space, ensure the program is running as expected, and supply the repl.it link in this form
  + Screen shots of the 3 matrix examples that are given below is required unless you have added file input as part of your program. Please put all three in the same google document file & make sure to share this with [gladbachj@umkc.edu](mailto:gladbachj@umkc.edu) and <stgdcg@umsystem.edu>.

**Rubric:**

This is the rubric used for grading [CS201 Program 2 - Matrix Functions](https://umkc.box.com/s/0eggb9bgqnzqp0l5mupned34zm2ab3c2)

**Sample Output:**

**Example 1 (from example given above):**

Calendar

Description automatically generated

Text

Description automatically generated**Example 2**

**(from matrix in data.txt)**

**NOTE: The input from the file will continue to display output for each matrix in the file.**

**Example 3**

Text

Description automatically generated

**Bonus Example (Route Checking)**

Text

Description automatically generated