CSE108 – Computer Programming Laboratory

(Spring 2021)

Lab #8

May 7, 2021.

Hand-in Policy: Via Teams. No late submissions will be accepted. File name that you submit should be as following: *StudentNo.c*
Collaboration Policy: No collaboration is permitted.
Grading: This lab will be graded on the scale of 100.

---

1. void print_matrix(matrix initial_matrix)

Write a function that takes a structure type named matrix as an argument. The matrix structure type should contain neither variable nor any data type other than a 2D double array representing an actual 3x3 matrix and a double variable indicating determinant of it. This function should be called in main(…) and print the matrix in a pretty format (centered entries with at most 4 digits after the decimal point, see the following example).

```
1.0000    0.9134    0.2785
0.9058    0.6324    0.5469
0.1270    0.0975    0.9575
```

void inverse_matrix(matrix* initial_matrix, matrix* inverted_matrix)

Implement this function which should be called in main(…) after print_matrix(…). First, the function has to call the following function and store its determinant to the related variable inside matrix structure type;

void determinant_of_matrix(matrix* initial_matrix)

Then, if the 3x3 matrix is invertible, the inverse_matrix function should find the inverse of the matrix and also its determinant (remember that inverse requires determinant to be calculated). The calculated values should be stored in the specified variables in inverted_matrix. Lastly, you should print the inverted matrix (if it is invertible) in main(…) by using print_matrix. In case of a non-invertible matrix, just print a proper error message in main(…). You can use any additional helper function you construct by yourself. All the formulas that you need as follows:

Determinant of A matrix $M = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ given by $|M| = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$.

The inverse of a matrix is given by $M^{-1} = \frac{1}{|M|}\begin{bmatrix} \begin{vmatrix} e & f \\ h & i \end{vmatrix} & -\begin{vmatrix} d & f \\ g & i \end{vmatrix} & \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ -\begin{vmatrix} b & c \\ h & i \end{vmatrix} & \begin{vmatrix} a & c \\ g & i \end{vmatrix} & -\begin{vmatrix} a & b \\ g & h \end{vmatrix} \\ \begin{vmatrix} b & c \\ e & f \end{vmatrix} & -\begin{vmatrix} a & c \\ d & f \end{vmatrix} & \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{bmatrix}^T$ where T indicates transpose. Obviously, the inverse is not defined if the matrix has zero determinant.

2. double find_orthogonal(vector vec_1, vector vec_2, vector* output_vec)

Write a function that takes the vector structure type as an argument. vector should solely contain x, y, z dimensions of 3D a vector as double numbers. This function should calculate and return the angle between input vectors vec_1 and vec_2 in degrees. It should find the vector orthogonal to the given two vectors (vector cross product) and return it in the output argument. The angle and the vector found should be displayed in main(…). You can utilize math.h library functions.

For two given vectors $\vec{A} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ and $\vec{B} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$, the angle between is given by $\theta = \cos^{-1}\frac{\vec{A}.\vec{B}}{\|\vec{A}\|\|\vec{B}\|}$ (where $\|\vec{A}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ and $\vec{A}.\vec{B} = a_x b_x + a_y b_y + a_z b_z$) and the cross product by $\vec{A}\ x\ \vec{B} = \begin{bmatrix} \begin{vmatrix} a_y & a_z \\ b_y & b_z \end{vmatrix} & -\begin{vmatrix} a_x & a_z \\ b_x & b_z \end{vmatrix} & \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} \end{bmatrix}^T$

3. polynomial get_integral(third_order_polynomial p1, third_order_polynomial p2, int a, int b)

Write a function that takes third_order_polynomial structure type as arguments. third_order_polynomial structure type should contain neither variable nor data type other than double variables representing coefficients of the third-degree polynomial (for $ax^3 \neq 0$). In this task, the program should ask for the user to input two third-degree polynomials and interval values. Between specified intervals[a, b], by calculating the integral of multiplication of these polynomials, it should return a new polynomial structure type containing coefficients (including the constant) and its value between [a, b] of the integrated polynomial. polynomial structure type should contain neither variable nor data type other than double variables representing coefficients of the integrated polynomial and a char variable indicating constant. In addition to this, it should also contain a double variable representing its value between [a, b]. Found coefficients and the value between [a, b] should be printed inside main(...).