BİL 214 – System Programming

Homework #3

Assigned on 04.10.2022 – Due on 11.10.2022

- Submit one C source file per question.
- If a source file fails to compile with the gcc compiler, you get zero credits for that question.
- Make sure your submission file names are formatted as:
  FirstName_LastName_StudentID_HW#_Q#.c
  For example: **Toygar_Akgun_123456789_HW3_Q1.c**
  Toygar_Akgun_123456789_HW3_Q2.c
  Toygar_Akgun_123456789_HW3_Q3.c …

1. [10 points] Write a C program that prompts the user to enter three floating point numbers and reads these inputs into three floats, say, A, B and C. Then, it does the following multiplications using these floats:

   D = (A * B) * C
   E = A * (B * C)

   Once these computations are done, your program evaluates the following condition:

   D == E

   and reports the result by printing one of the messages: “D is equal to E.” **-or-** “D is NOT equal to E.”.
   [Can you figure out some inputs that would lead to “D is equal to E.”?]
   [Can you figure out some inputs that would lead to “D is NOT equal to E.”.]

2. [15 points] An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3. Write a function **isPerfect** that determines whether parameter number is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and 1000. Print the factors of each perfect number to confirm that the number is indeed perfect.

3. [20 points] A prime integer is any integer greater than 1 that can be divided evenly only by itself and 1. The Sieve of Eratosthenes is a method of finding prime numbers. It works as follows:

   a) Create an array with all elements initialized to 1 (true). Array elements with prime indices will remain 1. All other array elements will eventually be set to zero.

   b) Starting with array index 2 (index 1 is not prime), every time an array element is found whose value is 1, loop through the remainder of the array and set to zero every element whose index is a multiple of the index for the element with value 1. For array index 2, all elements beyond 2 in the array that are multiples of 2 will be set to zero (indices 4, 6, 8, 10, and so on.). For array index 3, all elements beyond 3 in the array that are multiples of 3 will be set to zero (indices 6, 9, 12, 15, and so on.).

   When this process is complete, the array elements that are still set to 1 indicate that the index is a prime number. Write a program that uses an array of 1,000 elements to determine and print the prime numbers between 1 and 999. Ignore element 0 of the array.

4. [20 points] The Hadamard product (also known as the element-wise product, entry-wise product) is a binary operation that takes two matrices of the same dimensions and produces another matrix of the same dimension as the operands, where each element (i, j) of the output is the product of elements (i, j) of the two input matrices.

   Using two-dimensional arrays and assuming that all possible input matrices are smaller than 20x20, write a C program that implements the Hadamard product on **integer** matrices. Your code should take the input matrices from the user via terminal input, compute the Hadamard product of the inputs and print out the result. You are required to conduct dimensionality checks on the inputs and accept only matrices of equal sizes.

5. [35 points] A two-dimensional matrix can be multiplied by another matrix to give a matrix whose elements are the sum of the products of the elements within a row from the first matrix and the associated elements of a column of the second matrix. Both matrices should either be square matrices, or the number of columns of the first matrix should equal the number of rows of the second matrix.

   To calculate each element of the resultant matrix, multiply the first element of a given row from the first matrix and the first element of a given column in the second matrix, add that to the product of the second element of the same row and the second element of the same column, and keep doing so until the last elements of the row and column have been multiplied and added to the sum.

   Write a program to calculate the product of 2 **integer** matrices and store the result in a third matrix. Use two-dimensional arrays for your implementation. Assume that all possible input matrices are smaller than 20x20. Your code should take the input matrices from the user via terminal input, compute matrix product of the inputs and print out the result. You are required to conduct dimensionality checks on the inputs.