Programming Design Homework Assignment S8

Due: 2022/4/23 23:00

※Notice:

1. Please follow the rules for homework assignments announced on the course website.
2. The standard template library and unrelated macros are not permitted in this assignment.

There are three questions in this assignment, first one is programming, and others are short answer questions.

PART I Programming (60% in total)

- Programming formats：(12%)
  1. The structure of your program (3%)
  2. Clear and readable code layout (3%)
  3. Clear comments for understanding your program (3%)
  4. The copyright and short description of each question (3%)
- Question 1：(48%)

For this assignment, please use dynamic memory allocation to create two 2-dimensional arrays as matrices. The First matrix A is an m × n matrix and the second matrix B is a p × q matrix. The value of m, n, p and q are integer between 2 to 6 and generate by random function. The elements in the two matrices are also generated by random function (between -11.8 and +5.1). Rewrite HWS7-2 with several functions which are shown below.

1. Generate a integer number randomly (input upper and lower bond, and return a random number).
   inline int random_int(const int lower, const int upper);
2. Generate a floating number randomly (input upper and lower bond, and return a random number).
   inline float random_float(const float lower, const float upper);
3. Create a two 2-dimensional arrays.
   double create(const int rows, const int columns);**
4. Fill in elements of a matrix randomly. The function is a void function. Student shall deign the input parameters.
   void create_elements(……);
5. Do the Kronecker product of two matrices (A⊗B). The resulting matrix, and its rows and columns shall return to main function by using the first (double **& result), second (int& r_rows) and third (int& r_columns) parameters of this function, respectively.
   void kronecker_product(double **& result, int& r_rows, int& r_columns, const double

***const*const matrix1, const int rows1, const int columns1, const double *const*const matrix2, const int rows2, const int columns2);**

6. Find the minimum element and its subscript of a matrix. The subscript is marked with 1. Ex: the minimum value is at the position of [2][2], it should be showed as (3,3). Student shall deign your own input parameters and return type.
   ……. minimum(……);

7. Release a block of memory that you've allocated previously using the operator new.
   void release(double **&matrix, const int rows);

8. Create the output function that print a matrix orderly. Student shall deign the input parameters.
   void output_matrix(……);

In main function, you shall call the above function to calculate and output the following information on the console window.

1. The two original matrices.
2. The matrix after Kronecker product of two matrices (A⊗B).
3. Output the minimum element and its subscript of the above three matrices.
   The subscript is marked with 1. Ex: the minimum value is at the position of [2][2], it should be showed as (3,3).

DO NOT revise the parameters or return type of functions if it already specified. DO NOT create any other function in your program. Use "const" to protect reassignment of variables and parameters.

PART II Short answer questions (40% in total)

- Question 2: (30%)

What are the type (value, pointer, reference) of the passing parameters and return value of three functions (4.create_elements, 5.kronecker_product and 6.minimum) in Question 1? What is the reason or benefit for using that type in each function? Please answer the question in your words as detailed as possible and submit as a text file named "HWS8-Q2.txt".

- Question 3: (10%)

Please compare two functions below with different types of passing parameters which are shown below.

a. **void release(double **matrix, const int rows);**
b. **void release(double **&matrix, const int rows);**

What is the different between them and which one is more efficient during the execution, and why? Please answer the question in your words as detailed as possible and submit as a text file named "HWS8-Q3.txt".