National University
of computer and emerging sciences

LAB TASK 03- POINTERS & DYNAMIC MEMORY ALLOCATION

Instructions

Strictly follow the following instructions, otherwise your tasks would not be graded.

- Following **File Naming Convention** must be followed:

  ROLLNO_Section_LabNo_Task.cpp

- Do not submit compressed files or project folder, only code file submission is required.

Problem 1:

Write a function that takes two arguments as inputs: a pointer to integer p and size s. You are required to allocate memory for an array with s elements of type int using "new" operator. As your next step, randomly initialize all the elements of the arrays and then update each element value with its square, i.e. replace each element value with its square.

Problem 2:

(a):

In this task we consider ptr as a 2D matrix. Whereas, number of rows would be fixed but make number of columns variables. To do this, instead of using arrays of fixed column size allocated on stack, you will make each element of array to have nColumns (passed as argument to the function) elements and store them on heap using new operator. Next, initialize the values of these arrays randomly using for loop and ptr. Finally display the sum of each individual array.
You are not allowed to access the array using original variable name.

(b):

In the previous task we make number of columns of a 2D matrix variables. Here we will make number of rows variables as well. Thus we will have a way of defining a generic 2D matrix according to user choice. Now your goal is to write a function that receives three arguments: (i) an alias to a 2D pointer; (ii)number of rows; and (iii) number of columns; Now your goal is to first allocate the memory for rows and then for columns dynamically using new operator.

(c):

In this function, your goal is to write code for deallocating a dynamically allocated 2D matrix. Your function will receive three arguments: (i) a 2D pointer; (ii) number of rows; and (iii) number of columns. Complete the code to properly deallocate.

Problem 3:

In this question you will use the functions defined in the previous two tasks to create and compute sum of two nrows X ncols matrices, where nrows and ncols will be passed as arguments to the function. First define two 2D pointers call them matrixA and matrixB. Now call the function defined in above tasks to allocate the memory for both the matrices. Next randomly fill both the matrices and then store the sum of these two matrices in new created matrix matrixC. Finally, deallocate the memory of matrixA and matrixB using function from the previous task and return the matrixC from the function without deallocating.

Problem 4:

(a):

Now you are quite comfortable with 2D pointers. Here your goal is to define and allocate memory for 3D pointers.
Now your goal is to write a function that receives four arguments: (i) an alias to a 3D pointer; (ii) number of pages (or number of matrices); (iii) number of rows; and (iv) number of columns. Now your goal is to first allocate the memory for pages, rows and then for columns dynamically using new operator.

(b):

In this function, your goal is to write code for deallocating a dynamically allocated 3D matrix. Your function will receives four arguments: (i) a 3D pointer; (ii) number of pages; (ii) number of rows; and (iv) number of columns. Complete the code to properly deallocate the 3D array.

(c):

In this question you will use the functions defned in the previous two tasks to create and compute sum of two nrows X ncols matrices, where nrows and ncols will be passed as arguments to the function. First create a 3D pointer tdp and use the above defned function to create and allocate a 3D matrix with 3 pages (or 3 matrices) and nrows rows and ncols columns. Next randomly fill the first two matrices (at index 0 and index 1 of tdp) and then store the sum of these two matrices in third matrix (at index 2 of tdp).
You will return the result of sum of matrices from the function.