Lab 5

Thursday, October 19, 2023

You will get 1 point for attending the lab, making an effort to work on the problems, and **discussing your work with a lab instructor during the lab**.

If you finish during the lab, demonstrate your working code to a lab instructor and they will give you the full 2 points. If you do not finish during the lab, discuss what you've done with a lab instructor and they will give you an opportunity to finish outside of the lab and attend office hours to demonstrate your solution.

Practice Problems. Complete the following function that tests two 5×5 matrices for equality.

```
// matrix_eq(A, B) returns true exactly when the 5 x 5 matrices
//   A and B are equal
bool matrix_eq(const int A[5][5], const int B[5][5]);
```

Complete the following function that swaps two rows of a 5×5 matrix.

```
// matrix_swap(A, i, j) swaps the ith and jth row of matrix A
// requires: 0 <= i < 5 and 0 <= j < 5
//           A points to a 5 x 5 matrix that can be modified
void matrix_swap(int A[5][5], const int i, const int j);
```

In `main`, use `assert` to provide at least two tests for `matrix_eq`, one with a true return and one with a false return. Use `assert` and `matrix_eq` to provide at least one test of `matrix_swap`.

Assessment Problem. Use `matrix_swap` to write a function `sort_matrix` that sorts the rows of a matrix by the entries in its initial column. An example with a 5×5 matrix would be that

$$\begin{bmatrix} 100 & 200 & 300 & 400 & 500 \\ 5 & 5 & 5 & 5 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 10 & 9 & 8 & 7 & 6 \\ 3 & 1 & 4 & 1 & 5 \end{bmatrix} \quad \text{should become} \quad \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 1 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 10 & 9 & 8 & 7 & 6 \\ 100 & 200 & 300 & 400 & 500 \end{bmatrix}.$$

```
// sort_matrix(A, n) sorts the rows of the 5 x 5 matrix A by the entries in
//   the initial column
// requires: A points to a 5 x 5 matrix that can be modified
void sort_matrix(int A[5][5]);
```

You can use any sorting algorithm that you want. Run your code on the above example using the provided code on Brightspace. (You will need to complete implementations of `matrix_eq` and `matrix_swap` for it to compile.)