Aditya Saraf (sarafa) and Autumn Blackburn (kblack37)

CSE 333: HW #2 2D Array Library

Our 2D array library defines type Array2DData_t as a void*. If the client's data is no larger than a pointer, they may cast their data to this type. Otherwise, a pointer to the data is stored. We also defined type Array2DDataFreeFnPtr as a pointer to a function that accepts a void* and frees it. Our 2D array library also defines a struct* - Array2D. This struct represents the 2D array, containing a void pointer to the first element in the array and int rows and int cols fields. Instances of Array2D are pointers that point to this struct.

With that in mind, our 2D array library defines the following functions:

- Array2D Array2D_create(int rows, int cols)
- int Array2D_set(Array2D a, int row, int col, Array2DData_t data)
- int Array2D_swap(Array2D a, int r1, int c1, int r2, int c2)
- Array2DData_t Array2D_get(Array2D a, int row, int col)
- Int Array2D_destory(Array2D a, Array2DDataFreeFnPtr data_free_function)

As all of these interface methods are straightforward in purpose and implementation, we don't provide further explanation. For a more detailed look at their purpose and implementation, please see the Array2D.h and Array2D.c, respectively.