Data Structures and Algorithms - Lab 1: Dynamic Array

Implement in C++ the given container (ADT) using a given representation and a dynamic array as a data structure. You are not allowed to use the vector from STL or from any other library.

Obs:

- Since your implementation will use dynamic allocation, it is a good practice to implement a destructor, copy constructor and assignment operator as well (even if they are not on the interface).

- You are not allowed to use the functions memcpy and realloc, because it is not safe to use memcpy and realloc on memory that was allocated with new. Also, if the memory location contains objects, undefined behavior can occur. The implementation might still work with these functions, but it is not a good practice to use them.

- If you need auxiliary functions, fell free to add them to the interface of the ADT, but make them private.

3. ADT Matrix – represented as a sparse matrix, compressed sparse line representation using dynamic arrays.