**CSCI 135 – Software Design & Analysis I**
**Assignment 3**

This assignment gives you practice programming with the "new" and "delete" operators.

## Introduction

One problem with dynamic arrays is that once the array is created using the `new` operator, the size cannot be changed. For example, you might want to add or delete entries from the array similar to the behaviour of a vector.

## Assignment

This project asks you to create a class called `DynamicStringArray` which includes member functions that allow it to emulate the behaviour of a vector of strings. The class must have:

- A private member variable called `dynamicArray` which references a dynamic array of type string
- A private member variable called `size` which holds the number of entries in the array
- A default constructor which sets the size 0 and the `dynamicArray` to `NULL`
- A function to return `size`
- A function named `addEntry` that takes a string as input. The function should create a new dynamic array one element larger than `dynamicArray`, copy all the elements from `dynamicArray` into the new array, add the new string onto the end of the new array, increment size, delete the old `dynamicArray` and then set `dynamicArray` to the new array
- A function named `deleteEntry` which takes a string as input. The function must search `dynamicArray` for the string. If not found, it returns false. If found, it creates a new dynamic array, one element smaller than `dynamicArray`. If must copy all elements except the input string into the new array, delete `dynamicArray`, decrement size and return true. For simplicity, if there are two strings with the same value, delete only the first value
- A function named `getEntry` which takes an integer as input and returns the string at that index in `dynamicArray`. It must return `NULL` if the index is out of the range of `dynamicArray`
- A destructor which frees the memory allocated to the dynamic array

Embed your class in a suitable test program.