Problem 1

Not using class **string** from the Standard Library (header ***string***), write your own class **String**, objects of which represents strings (arrays of characters) of arbitrary length. You may use functions from header ***cstring*** like **strlen**, **strcpy**, **strcat** or **strcmp** and also functions operating on single characters (like **toupper** or **tolower**) from header ***cctype***.

Define a constructor taking array of characters as well as all five "special members" (copy and move constructors, copy and move assignment operators and destructor). Define also methods and friend functions, in particular overloading operators of addition (concatenation) of strings and comparing them. They should also work when one of the arguments is a plain C-string.

Scheme of the program

download *MyString.cpp*

```
#include <iostream>
#include <cstring>   // strlen, strcpy, strcat, strcmp...
#include <cctype>    // tolower, toupper

// ...

int main() {
    String s = ("To " + String("be ") +
                "or not to be").toUpper() + "!";
    std::cout << s << std::endl;
    String a("a"), A = "A";
    std::cout << std::boolalpha
        << ("a" ==  a  && "a" !=  A  && a != A &&
            a  == "a" &&  A  != "a" && a == A.toLower() &&
            s.length() == 19) << std::endl;
    a = String(a).toUpper() + "li" + String("ce");
    std::cout << a << ", len=" << a.length() << std::endl;
}
```

Methods **toLower** and **toUpper** (changing the case of characters in a string) modify and return by reference the object they were invoked on.
Method **length** returns the number of characters in the string represented by the object (not counting the '\0' character at the end).

The program above, after implementing all functions, should print

```
TO BE OR NOT TO BE!
true
Alice, len=5
```