POO (16:00-17:00)

Write the class **CircularString** (that contains a fixed-sized string) so that the following code

```
int main()
{
    CircularString c('*', 5);
    cout << (const char*)c << endl;
    ((c += 'a') += 'b') += 'c';
    cout << (const char*)c << endl;
    c++;
    cout << (const char*)c << endl;
    c++; c++; c++;
    cout << (const char*)c << endl;
    --c;
    cout << (const char*)c << endl;
    int len = (int)c;
    for (int tr = 0; tr < len; tr++) { cout << c[tr] << ","; }
    cout << endl << "Len = " << len << endl;
    cout << "Ascii codes: ";
    for (int tr = 0; tr < 10; tr++) { cout << c(tr) << ","; }
    return 0;
}
```

compiles and upon execution prints the following to the screen:

```
*****
cba**
*cba*
ba**c
a**cb
a,*,*,c,b,
Len = 5
Ascii codes: 97,42,42,99,98,0,0,0,0,0,
```

Carefully read the main function to deduce what methods/operators should be included in CircularString class.

Constraints:

- You are not allowed to use **STL** at all (for vectors, strings, maps or any **template/object** defined in STL). The only exception is the usage of "**std::cout**" from the main function
- You are not allowed to use string manipulation functions defined in "string.h" such as **strlen**, **strcpy**, **strdup, strtok, strcmp**, etc, or string to number conversions.
- If you don't respect the previous conditions (e.g. use strlen, or strcpy, etc) → we will compute the correctness of the code, but the final grade will be half of the computed score for each particular code that uses those functions.

Observations:

- The constructor receives the size of the string (that will always be bigger than 1) and an initial character (that will always be different than 0) to fill out the string. In our case, `CircularString c('*', 5)` will create a string of fixed size 5 filled out with the asterisk character.

Grading (informative):

G1	Circular string destructor	1p
G2	Circular string constructor	3p
G3	Organize your project in 3 files: **main.cpp**, **CircularString.h** and **CircularString.cpp**	1p
G4	Organize your class **CircularString** to include private and public members, the definition of a constructor, and at least one operator. The private data should include one pointer and the fixed size of the string.	2p
G5	Operator++ to rotate the string to the right	4p
G6	Operator-- to rotate the string to the left	4p
G7	Operator+= to add a character on from of the string and remove the last one	4p
G8	Operator to allow the cast to a const char *	2p
G9	Operator to allow the cast to int (providing the size of the string)	2p
G10	Operator() to get the ASCII code (int value) from a position or 0 if out of bounds	2p
G11	Operator[] to get the character from a position or 0 if out of bounds	2p
G12	The program compiles and upon execution produces the expected results	3p