Lab 5. Operators

Write a class in C++ that has the following definition:

```
class Number
{
   // add data members
public:
   Number(const char * value, int base); // where base is between
2 and 16
   ~Number();

   // add operators and copy/move constructor

   void SwitchBase(int newBase);
   void Print();
   int  GetDigitsCount(); // returns the number of digits for the
current number
   int  GetBase(); // returns the current base
}
```

Organize the code in the following way:

- a header file called **Number.h**
- a cpp file called **Number.cpp** that contains the source code for class **Number**
- a main file called **main.cpp** that contains the main function and has an example on how to use **Number**. The example must include using all methods from the class.
- add the following operators: addition, subtraction, negate, index operator, relation operators (> , < , >= , <=, ==, etc)
- add copy & move constructors and move assignment operator
- when performing operations with two **Number** object that have a different base, the result (except for the relation and index operators) will have the biggest base of the two Number instances.
- for addition and subtraction use **friend** functions
- implement the -- operator with the following syntax: if used in a prefix form it will remove the first (most significant digit) from the number; if used in a post-fix form it will remove the last (least significant) digit from the number;

Example:

```
int main()
{
    Number n1("10110010",2);
   Number n2("734",8);
   Number n3("FF",16);

   printf("n1 has %d digits and it is written in base
%d\n",n1.GetDigitsCount(),n1.GetBase());
   for (int tr=0;tr<n1.GetDigitsCount();tr++)
   {
      printf("n1[%d]=%c\n",tr,n1[tr]);
   }
   n1.Print();
   n2.Print();
   n1 = (n2+n3-n1)+n1; // after this n1 will be in base 16
   n1.SwitchBase(2);
   n1.Print();

   if (n1>n2) printf("n1 is bigger than n2\n"); else printf("n2 is bigger than
n1\n");

   Number n4=12345; // n4 will be in base 10
   n1 = 255; // n1 will be 11111111 (value 255 from base 10 in base 2)
   n4 += n1;
   n4.Print();

   n4 = "13579"; // n4 mentains its base (10) and will be 13579
   n4.Print();
   --n4; // the first digit from n4 will be remove ==> n4 becomes 3579
   n4.Print();
   n4--; // the last digit from n4 will be remove ==> n4 becomes 357
   n4.Print();

   return 0;
}
```