

PIC 10B SPRING 2013 HOMEWORK 7

Assignment

Create a `BinaryInteger` class that mimics the way integers are stored in a computer, but which represents negative numbers in the ‘sign and magnitude’ form (i.e., the binary representation for -19 would be the same as 19 but with the signed bit ‘1’). It should have a private member variable which is a vector of boolean values, and another boolean which determines whether the integer is positive or negative.

Define the operators `+`, `-`, `&`, `|`, `^`, `~`, `<<`, `>>`, `<=<`, `>=>` to perform the usual operations on `int` data types. The main difference is that there is no inherent limit on the number of bits in your vector of booleans, so for example the bit shift operator shifting up can work indefinitely until your computer runs out of memory, and the interpretation of the bits for negative numbers will be different.

There should be

- `+` : `BinaryInteger` \times `BinaryInteger` \rightarrow `BinaryInteger` performs standard integer addition.
- `-` : `BinaryInteger` \times `BinaryInteger` \rightarrow `BinaryInteger` performs standard integer subtraction.
- `&` : `BinaryInteger` \times `BinaryInteger` \rightarrow `BinaryInteger` performs bit-wise AND operation.
- `|` : `BinaryInteger` \times `BinaryInteger` \rightarrow `BinaryInteger` performs bit-wise OR operation.
- `^` : `BinaryInteger` \times `BinaryInteger` \rightarrow `BinaryInteger` performs bit-wise XOR operation.
- `BinaryInteger~` : `BinaryInteger` \rightarrow `BinaryInteger` performs bit-wise negation.

Your `+` and `-` functions should perform bit-wise addition and subtraction on the vector of bools. Do NOT simply convert the vector of bools to an integer, have C++ perform the integer arithmetic, and then convert it back to binary. The whole point of the assignment is that these operations are done bit-wise, which frees them from the constraint of being between the min and max values the integer data type can hold.

Also overload the operators `<<` and `>>` in TWO different ways. One way is the input/output, which are non-member functions (feel free to have `BinaryInteger` grant them friendship status!), so

- `<<`: `stream` \times `BinaryInteger` \rightarrow `stream` outputs the integer value of the binary vector.
- `>>`: `istream` \times `BinaryInteger` \rightarrow `istream` accepts an integer value and then converts it to a binary representation to store it.

and the second way is the bit-shift operations, which are member functions

- `BinaryInteger <<`: `int` \rightarrow `BinaryInteger`
- `BinaryInteger >>`: `int` \rightarrow `BinaryInteger`

- `BinaryInteger <<=: int → BinaryInteger`
- `BinaryInteger >>=: int → BinaryInteger`

Also implement the following

- A Binary print function that prints the vector of bools as a vector of 0s and 1s, and prepends a '-' symbol if it is negative
- A default constructor that initializes a vector of bools and sets sign to 0,
- a constructor with one parameter of int type, and converts it to binary
- a constructor that takes a vector of booleans and a bool for the sign as the binary representation and sign.

Finally, implement a *private* function called `CleanUp` that eliminates any unnecessary 0s in the binary expansion, so for example after you run some calculations you may have a vector of bools with entries (0,0,0,1,1,0,1), and after running `CleanUp` on the vector it will be (1,1,0,1).

Place your code in a source file labeled `hw7.cpp`. ***If your file is not named this exactly, your homework will not be collected.*** As with all programs in this course, your code should contain useful comments. In particular, your name, the date, and a brief description of what the program does should appear at the top of your source file.

What to Turn in

Place in your Submit folder the source file `hw7.cpp` with exactly this name (all lowercase, no spaces). The files will be automatically collected on Friday 5/17/13 at 5:00pm.

Grading		
Correctness	No errors, input/output correct, output presented nicely	5 points
Binary	Correctly implements Binary operations	10 points
Style	Variable names, comments, indentation	5 points
	TOTAL	20 points

Note on grading: There is an automatic 5 point penalty for any homework that does not compile.

```

Please input an integer n: 19
The binary representation for n is: (1,0,0,1,1)
Please input an integer m: -37
The binary representation for m is: -(1,0,0,1,0,1)
Please input binary representation, first bit indicates sign (enter -1 when finished):
1
1
0
0
1
1
0
-1
Your integer value for r is: -38
n = 19 = (1,0,0,1,1)
~n = -12 = -(1,1,0,0)
n+m = -18 = -(1,0,0,1,0)
n&m = 1 = (1)
n^r = -53 = -(1,1,0,1,0,1)
2r = -76 = -(1,0,0,1,1,0,0)
n-m = 56 = (1,1,1,0,0,0)
n | m = -55 = -(1,1,0,1,1,1)
256*n = 4864 = (1,0,0,1,1,0,0,0,0,0,0,0,0)
n/8 = 2 = (1,0)

```