CSCE 240 HOMEWORK 1 DUE: 04 FEB 2020

You shall submit a zipped, **and only zipped**, archive of your homework directory, hw1. The directory shall contain, at a minimum, the file `binary_converter.cc`. Your submission file must be named hw1.zip.

I will use my own makefile to make your `binary_converter.cc` file. Do not use a header for this assignment. My grader will not look for one.

Everyone in computing should have a basic understanding of binary integers. To that end, you will write an app to convert decimal to binary.

The application should perform as follows:

- The program is called and runs with no prompt for input.
- The first value provided to the standard input stream shall be an integer and be interpreted as the number of bits of binary output. This value should be something in the range [1, 31]. Do be careful, sometimes I get careless and enter values outside this range. If that happens, the app should quietly exit with a value of 1.
- The second value provided is the decimal integer to be converted. Though I can guarantee that you will see values in the range $[0,\ 2^{31} - 1]$, I am sometimes careless about the number of bits required to represent a decimal integer. If I make a mistake by entering a value too large for the number of bits, that is on me and you should still only print the number of bits requested. If I enter a negative value, do not print a two's complement; again, quietly exit, this time with a value of 2.
- An algorithm to convert decimal number, n, to binary (without arrays):

```
1. Use loop to calculate the largest power of 2, call it $i$, in $n$, print a 1 and subtract $2^i$ from $n$.[1]
2. For each exponent, $j$, from $i - 1$ down to 0,
   (a) if that $2^j \leq n$, print a 1 and subtract $2^j$ from $n$
   (b) otherwise if that $n < 2^j$, print a 0.
```

  There are other algorithms which are easier as well as improvements to this algorithm. You should not solve this with an array (or string). That defeats the purpose of the assignment.

I provided you with a test file to test your code. You should ensure that your code satisfies the tester's requirements. It is the tool I will use to grade your submissions. I will only change the input and expected values.
To utilize the tester, you will need access to a python3 interpreter. The tester can be called using the make file, assuming that `python3` is in your path and that your present working directory is `../hw1`

Point Awards

Correctly converting and printing a valid value is worth 2 points. Correctly catching an incorrect number of bits is worth $\frac{1}{2}$ points. Correctly catching negative decimal input is worth $\frac{1}{2}$ points. Style is worth 2 points. I have provided you a make file. You should definitely read the makefile and I would encourage you to read the python tester.

[1] Do not forget to check whether this exponent is less than or equal to the first integer passed to standard input. If it is outside that value, you should subtract as indicated, but not print. You should then find the next largest exponent and try again.

Late assignments will lose 25% per day late, with no assignment begin accepted after 4 days (100% reduction in points).

Check your syllabus for further caveats of grading.

The End.