Files to submit: div.c
Time it took Matthew to complete: 10 mins

- All programs must compile without warnings when using the -Wall and -Werror options
- Submit only the files requested
  - Do **NOT** submit folders or compressed files such as .zip, .rar, .tar, .targz, etc
- If submitting in a group on Grade Scope please make sure to mark your partner.
  - Only one of you has to submit there
- Your program must match the output exactly to receive credit.
  - Make sure that all prompts and output match mine exactly.
  - Easiest way to do this is to copy and paste them
- All input will be valid unless stated otherwise
- Print all real numbers to two decimal places unless otherwise stated
- The examples provided in the prompts do not represent all possible input you can receive.
- All inputs in the examples in the prompt are underlined
  - You don't have to make anything underlined it is just there to help you differentiate between what you are supposed to print and what is being given to your program
- If you have questions please post them on Piazza

1. Write a C program called **div.c** that implements division on two unsigned integers.
   1. Name your executable **div.out**
   2. You cannot use division in your solution
   3. Arguments should be accepted from the command line
      1. The first argument is the dividend
      2. The second argument is divisor
   4. Your program should display the quotient and remainder after doing the division
   5. Your program must complete in O(1) (constant) time
      1. This is possible because an integer is 32 bits long and so the loop that does the division should not take longer than 32 iterations.
      2. Because of this restriction the following solution is not acceptable as it does not meet the O requirements

         ```
         void bad_div(unsigned int dividend,
         unsigned int divisor,
         unsigned int* quotient,
         unsigned int *remainder){
           *quotient = 0;

           while(dividend >= divisor){
             (*quotient)++;
             dividend -= divisor;
           }

           *remainder = dividend;

         }
         ```

      3. In order to meet the O requirements you will have to division in base 2 as you would by hand. See these 2 articles for some examples: Dr. Math and Exploring Binary
         1. Hint: use bitwise operators
         2. The one step that they leave out and one that you normally skip when doing division by hand is checking to see how many times the divisor goes into the dividend for the numbers that contain fewer digits than the divisor.
            1. For example: 30 / 15
            2. First we should check how many times does 15 go into 3. The answer is 0.
            3. Then we check how many times does 15 go into 30, which is 2.
            4. So our answer would be 02 R 0
      4. Examples:

         ```
         1. ./div.out 10 5
            10 / 5 = 2 R 0
         2. ./div.out 100 17
            100 / 17 = 5 R 15
         ```