**Timing Recursion**

Your program should be able to replicate this output:

**>>>**

**Please enter an upper bound (int): cat**

**That's not an integer.**

**Please enter an upper bound (int): 15**

**Please enter FIB for fibonacci or FACT for factorial: no**

**Please enter FIB for fibonacci or FACT for factorial: fib**

**Calculating fibonacci(1) = 1 took: 2.83395503222e-06 seconds.**

**Calculating fibonacci(2) = 2 took: 2.47971065319e-06 seconds.**

**Calculating fibonacci(3) = 3 took: 2.83395503221e-06 seconds.**

**Calculating fibonacci(4) = 5 took: 3.54244379027e-06 seconds.**

**Calculating fibonacci(5) = 8 took: 4.95942130638e-06 seconds.**

**Calculating fibonacci(6) = 13 took: 6.02215444347e-06 seconds.**

**Calculating fibonacci(7) = 21 took: 8.50186509666e-06 seconds.**

**Calculating fibonacci(8) = 34 took: 1.2398553266e-05 seconds.**

**Calculating fibonacci(9) = 55 took: 1.84207077094e-05 seconds.**

**Calculating fibonacci(10) = 89 took: 2.79853059432e-05 seconds.**

**Calculating fibonacci(11) = 144 took: 4.39263029994e-05 seconds.**

**Calculating fibonacci(12) = 233 took: 6.90776539103e-05 seconds.**

**Calculating fibonacci(13) = 377 took: 0.000110170001878 seconds.**

**Calculating fibonacci(14) = 610 took: 0.000175350967619 seconds.**

**Calculating fibonacci(15) = 987 took: 0.000281270036948 seconds.**

**Total time: 0.145071929321 seconds.**

As well as

**>>>**

**Please enter an upper bound (int): 20**

**Please enter FIB for fibonacci or FACT for factorial: fact**

**Calculating factorial(1) = 1 took: 2.83395503222e-06 seconds.**

**Calculating factorial(2) = 2 took: 5.31366568541e-06 seconds.**

**Calculating factorial(3) = 6 took: 2.47971065319e-06 seconds.**

**Calculating factorial(4) = 24 took: 3.18819941125e-06 seconds.**

**Calculating factorial(5) = 120 took: 3.18819941125e-06 seconds.**

**Calculating factorial(6) = 720 took: 3.54244379026e-06 seconds.**

**Calculating factorial(7) = 5040 took: 3.54244379028e-06 seconds.**

**Calculating factorial(8) = 40320 took: 3.8966881693e-06 seconds.**

**Calculating factorial(9) = 362880 took: 3.54244379028e-06 seconds.**

**Calculating factorial(10) = 3628800 took: 1.84207077094e-05 seconds.**

**Calculating factorial(11) = 39916800 took: 4.60517692735e-06 seconds.**

**Calculating factorial(12) = 479001600 took: 2.12546627416e-05 seconds.**

**Calculating factorial(13) = 6227020800 took: 1.45240195401e-05 seconds.**

**Calculating factorial(14) = 87178291200 took: 6.73064320153e-06 seconds.**

**Calculating factorial(15) = 1307674368000 took: 2.40886177739e-05 seconds.**

**Calculating factorial(16) = 20922789888000 took: 2.37343733948e-05 seconds.**

**Calculating factorial(17) = 355687428096000 took: 2.90480390802e-05 seconds.**

**Calculating factorial(18) = 6402373705728000 took: 2.30258846368e-05 seconds.**

**Calculating factorial(19) = 121645100408832000 took: 2.37343733948e-05 seconds.**

**Calculating factorial(20) = 2432902008176640000 took: 8.50186509666e-06 seconds.**

**Total time: 0.176302822265 seconds.**

Your mission will be to do the following:

1. Copy the factorial and fibonacci functions. Get a number from the integer as an upper bound. Validate the integer.

2. Ask the user to enter “FIB” for fibonacci or “FACT” for factorial. Keep asking until they enter one of those (ignore case).

3. If the user asks for fibonacci, compute the fibonacci numbers from 1 to their upper bound, inclusive. If they ask for factorial, compute those values instead.

4. Compute the amount of time it takes to calculate each value, and print this out.

5. Compute the total amount of time it takes to calculate all values, and print this out.