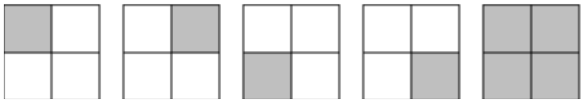
CMPS 2433 Discrete Structures and Analysis
Program 2 Fall 2021 **Due Wednesday, October 20, 2021**

PURPOSE: To reinforce the concepts of summing a sequence and solving recurrence relations, along with comparing/contrasting recursive and non-recursive solutions to problems.

PROBLEM: Richard Feynman was a well-known American physicist and a recipient of the Nobel Prize in Physics. He worked in theoretical physics and pioneered the field of quantum computing. He visited South America for ten months, giving lectures and enjoying life in the tropics. His life-long addiction was solving & making puzzles, locks, and cyphers. Recently, an old farmer who was a host to the young physicist in 1949, found some papers and notes that is believed to have belonged to Feynman. Among the notes, there was a napkin on which a simple puzzle was written: "How many different squares are there in a grid of *N X N* squares?" On the same napkin there was a drawing that is reproduced below, showing that, for *N* = 2, the answer is 5.



Your program must have THREE different functions to find the number of squares in an NXN grid. One of the functions must be iterative using a loop (summation of a sequence), one must be recursive, and the third must use a closed form solution, that is a one-line formula.

INPUT: The input contains M test cases, where M is on the first line of input. Each test case is composed of a single line, containing only one integer *N*, representing the number of squares in each side of the grid ($1 \leq N \leq 100$)

Input file: Feynman.dat

Sample input file:

```
5
10
1
4
7
14
```

OUTPUT: Your name as first line of output, another line has the column headings shown below. The N lines after have for each N, the number of squares using the iterative solution, the number of squares using the recursive solution and the result of the closed form solution. Format output with iomanip.h.

Sample corresponding output:

```
Catherine Stringfellow
Program 2: Squares

 N   LoopSum       RecSoln      Closed
10     385          385          385
 1       1            1            1
 2       5            5            5
 3      14           14           14
 7     140          140          140
14    1015         1015         1015
```

Hints:

1) Solve the problem in pieces – don't procrastinate!
2) Get a program to work that opens a file and just prints out the headings and the N values.
3) Next, solve the problem for the closed form solution, by computing the results for N=1, 2, 3, 4, 5 by hand so you can see the pattern. Turn in these hand calculations in *with* the hand calculations for your input.
4) Next, code the iterative solution. You will need that starts at 1, and you will be raising your loop control variable to a power.
5) Finally, solve the problem recursively. The recursive solution is based on the previous value + some value to a power.
6) LOOK at the Rubric, so you don't miss points.

Turn in:

- Upload the .cpp file to the dropbox in D2L by 9am on the due date.
- In class, turn in
    - Printout of your program with
    - Hand caculations for N = 1,2,3,4,5 and your data.
    - Printout of your input file
    - Printout of two output files, corresponding to my input and your input files.