More loops
Ch 3.3-3.4

```c
#include <stdio.h>
int main(void)
{
    int count;
    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```
Review: Loops

We put a loop around code that we want to run more than once.

If we have an easy sequence \((0, 1, 2, \ldots, 10)\) of values we want to go over, the for loop is nice.

Otherwise, the while loop is a bit more general and is typically more useful if we are asking the user to control the loop.
Write a program that asks the user to input a value, then show the sum from 1 to that value in the following format:

Find the sum from 1 to what value?  5

1+2+3+4+5  =  15

(See: sumToN.cpp)
Nested for loop

Now modify the code so it shows all sums less than or equal to the entered values, as such:

Find the sum from 1 to what value? 4
1 = 1
1+2 = 3
1+2+3 = 6
1+2+3+4 = 10
(See: sumAllToN.cpp)
Nested for loop

Like nested if statements, we can also make nested loops (which can cause headaches)

It might help to think of each loop as an added dimensions:

1 loop = 1 dimension (line/ruler)
2 loops = 2 dimensions (plane/square/area)
3 loops = 3 dimensions (volume/cube)
...

(See: nestedLoop.cpp)
Ask the user for a size of matrix, then show the identity matrix for that dimension:

What size? 4

1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1

(See: identityMatrix.cpp)
while vs do-while

(see: vendingV2.cpp)
Overview

Peripheral
- file I/O
- op. overload

Advanced
- recursion
- pointers
- dynamic memory

Very Useful
- scope
- array
- string
- classes

Useful
- loop
- types
- if/else
- ops

Essentials
- functions