

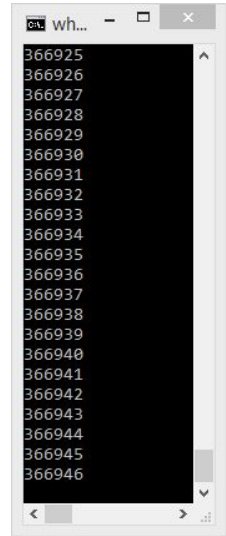
Implement a structure

- Implement a structure to fully define a **chicken**
 - Don't waste too much time on the details
- Bonus Points:
 - Keep track of a coop (many) of chickens
 - Keep track of each chicken's parents (Be creative)



Learning to Count

- Use the while loop to indefinitely display a series of increasing numbers
- Start from 1 and count upward
- It should continue counting upward as long as the program is running
- I don't care about overflow



A screenshot of a terminal window titled "wh...". The window displays a vertical list of numbers starting from 366925 and increasing by 1 up to 366946. The numbers are printed in a monospaced font on a black background. The window has standard OS controls (minimize, maximize, close) at the top and a scrollbar on the right side.

```
366925
366926
366927
366928
366929
366930
366931
366932
366933
366934
366935
366936
366937
366938
366939
366940
366941
366942
366943
366944
366945
366946
```

while (user==gullible)

Write a program that continues to ask the user to enter any number other than 5 until the user enters the number 5

1. After 10 iterations if the user still hasn't entered 5, tell the user "Wow, you're more patient than I am. You win." and exit

Even

Output all even numbers from A, to B.

Inputs (*with cin*): int A, int B

Even but not $k \cdot M$

Output all even numbers from A, to B, **except** the multiples of M.

Inputs (*with cin*): int A, int B, int M

SumAvgMin

Read N decimal numbers from the user.

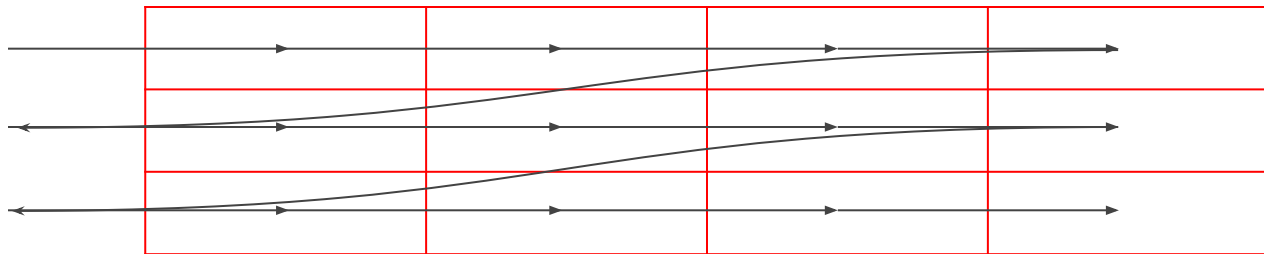
1. Output their sum.
2. Output the mean (average).
3. Output the minimum number.

Inputs (*with cin*): int N, <N more numbers of type **double**>

Do **not** use an array.

Iterating through n-dimensional arrays

- 2D: Column by column and then row by row
- 3D: Column by column, row by row, and then layer by layer



Hint: You can place a for loop inside
another for loop.

Pizza Glutton

Hint: You can use an integer variable to declare an array!
`cin>>n; int array[n];`

Write a program that:

1. Asks how many people ate pizza
2. Asks how many slices each one ate
3. Finds which person ate the most slices
4. Tells you the fraction of an 8 slice pizza each one had
 - a. X slices is X/8 ths of that pizza
 - b. Careful to use the correct type (`int` is not correct)
5. Formats “pizza” in plural when more than 1 pizza

```
How many people ate pizza? 3
Slices person 1 ate: 4
Slices person 2 ate: 3
Slices person 3 ate: 93
Person 3 ate the most!
Person 1 ate 0.5 pizza.
Person 2 ate 0.375 pizza.
Person 3 ate 11.625 pizzas.
```


Triangle

Write a **void** function that, given a number N, outputs a “triangle” of asterisks to the screen.

15→

```
* * * * *
```

```
    * * * * *
```

```
      * * * * *
```

```
        * * * * *
```

```
          * * * * *
```

```
            * * * * *
```

```
              * * * * *
```

```
                * * * * *
```

```
                  *
```

12→

Even Function

Write a function that returns true if a number is even.

Parameters: int number

Returns: a boolean (true or false)

Closest Even

Write a function that returns the closest even number to N.

13→14 16→16

Parameters: int N

Returns: an integer

Factorial

Write a function that returns **N!**.

$$9! = 9*8*7*6*5*4*3*2*1 = \underline{362880}$$

$$N! = N*(N-1)*(N-2)*(N-3)*...*3*2*1$$

Parameters: int N

Returns: an integer

Prime

Write a function that returns true if a number is a prime.

Parameters: int number

Returns: a boolean (true or false)

Goldbach's Conjecture

Given an integer N , find 2 prime numbers A , B whose sum equals N .

$$N=A+B$$

Inputs (*cin*): N

Outputs: A , B


Area

Write a function to calculate the area of a triangle

Parameters: Length of the sides (double a, double b, double c)

Return: The area of the triangle

Hint: Heron's formula: $A = \sqrt{s(s-a)(s-b)(s-c)}$, $s = \frac{a+b+c}{2}$.

If you don't know how to find the square root,  it

Knapsack with Rocks

Rocks are given in ascending density.

There are N rocks. Each one has mass and volume.
You are allowed to break rocks into pieces.

You have a knapsack that holds up to V volume.

What's the maximum mass the knapsack can have when perfectly filled with rocks?

Inputs (*cin*): N , V , <the mass and volume of N rocks>