More examples w/ loops.

"<u>Brute force</u>" for finding GCDs.

gcd(a,b) = largest integer d s.t. d|a <u>and</u> d|b.

"such that"

(Notation: $d|a$ means $\exists\, c \in \mathbb{Z}$ s.t. $d \cdot c = a$.)

"there exists"

Ex: gcd(12, 18) = 6.

Brute force: enumerate candidate solutions; test each one until solution is found.

for gcd(a,b), candidate solutions = 1, 2, ..., min(a,b)

$\longleftarrow$

In c++:

```
    int m = a;
    if(b < a) {
        m = b;
    }
    // now m = min(a,b)
  int d = m; // candidate gcd
  while (d >= 1) {
      // test if d is a common divisor:
      if (a%d == 0 && b%d==0) {
          cout << d << " = gcd";
          break;
      }
      d--; // same as d = d-1;
  }
```

Note: there's another flavor of loop: the "for loop".

```
for (initialization; boolean expr; update;) {
        // code...
}
```

one time only. after body of loop.

```
E.g.  for (int d = n; d >= 1; d--) {
          if (a%d == 0 && b%d == 0) {
              cout << d << " = gcd";
              break;
          }
      }
```

Example: on input of an integer n, find the exponent of the largest power of 2 that divides n.

e.g. if $n = 40 = 2^3 \cdot 5$, answer would be 3

or if $n = 7$, answer would be 0.

Idea: "remove" 2's by division and count.

```
  int n;
  cin >> n;
 int count = 0;
 while (n % 2 == 0) {
       n = n/2;  // shorthand: n/= 2.
       count ++;
 }
 cout << count;
```