Count factors of a number x in three different ways:

```
int countFactors(int x) {
    int count = 0;
    for (int i = 1; i <= x; i++) {
        if (x % i == 0) {
            count++;
        }
    }
    return count;
}


int countFactors(int x) {
    int count = 0;
    for (int i = 1; i <= x/2; i++) {
        if (x % i == 0) {
            count++;
        }
    }
    return count;
}

int count = 0;
    for (int i = 1; i <= sqrt(x); i++) {
        if (x % i == 0) {
            if (i == x / i) {
                ++count; // i and x/i are the same, count only once
            } else {
                count += 2; // count both i and x/i
            }
        }
    }
    return count;
```

Sieve of Eratosthenes:

```
#include <iostream>
#include <vector>

void sieveOfEratosthenes(int n) {
    // Create a boolean vector and initialize all entries as true.
    // A value in prime[i] will be false if i is not a prime, true otherwise.
    std::vector<bool> prime(n + 1, true);
    prime[0] = prime[1] = false; // 0 and 1 are not prime numbers

    for (int i = 2; i * i <= n; i++) {
        // If prime[p] is not changed, then it is a prime
```

```
        if (prime[p]) {
            // Update all multiples of p to not prime
            for (int j = i * i; j <= n; j += p) {
                prime[i] = false;
            }
        }
    }

    // Print all prime numbers
    for (int p = 2; p <= n; ++p) {
        if (prime[p]) {
            std::cout << p << " ";
        }
    }
    std::cout << std::endl;
}

int main() {
    int n = 30; // Change this value to find all primes up to n
    std::cout << "Prime numbers up to " << n << " are: ";
    sieveOfEratosthenes(n);

    return 0;
}
```

Greatest Common Divisor:

```
int gcd = 1; // Initialize gcd to 1

    // Iterate from 1 to min_val to find the greatest common divisor
    for (int i = 1; i <= min(a,b); i++) {
        if (a % i == 0 && b % i == 0) {
            gcd = i;
        }
    }

    return gcd;


int gcd = 1; // Initialize gcd to 1

    // Iterate from 1 to min_val to find the greatest common divisor
    for (int i = min(a,b); i>=1; i–) {
        if (a % i == 0 && b % i == 0) {
            gcd = i;
            break;
        }
    }
```

```
return gcd;
```