

最大公因数

1.自己手写（了解一下gcd实现原理）

理论：欧几里得算法（辗转相除法）： $\forall a, b \in N, b \neq 0$, 有 $gcd(a, b) = gcd(b, a \% b)$.

对比：更相减损术在某些情况下效率低，不用。

```
int gcd(int a, int b)
{
    if(b==0) return a;
    return gcd(b, a%b);
}
//或:
int gcd(int a, int b){
    return b==0 ? a : gcd(b, a%b);
}
```

2.直接调用库函数（常用）

直接使用C语言的库函数 `_gcd(a,b)`

注意点：

- 1.参数为int/long long，返回值为int/long long。不能用于浮点型。
- 2.两个参数的类型必须要相同。

最小公倍数

`lcm=a*b/_gcd(a,b)`

质数判断

1.试除法（时间复杂度： \sqrt{N} ）

```
bool isprime(int n){
    if(n<=1) return false;
    for(int i=2; i*i<=n; ++i)
        if(n%i==0) return false;
    return true;
}
```

2.更快的方法

```
bool isprime(int n){
    if(n<=1) return false;
    if(n==2 || n==3) return true;
    if(n%6!=1&& n%6!=5) return false;
    for(int i=5; i*i<=n; i+=6)
        if(n%i==0 || n%(i+2)==0) return false;
    return true;
}
```

sqrt()函数使用

参考资料: <https://cplusplus.com/reference/cmath/sqrt/>

常用: `double sqrt(double x);`

pow()函数使用

参考资料: <https://cplusplus.com/reference/cmath/pow/>

常用: `double pow (double base, double exponent);`

对比: pow可以计算比如 $8.3^{0.3}$ 这样的数, 但快速幂不行。

快速幂: $a^b, a \in N^+, b \in N$

不需要取模:

```
int qpow(int a,int b){
    int ans=1;
    while(b){
        if(b&1) ans*=a;
        a*=a;
        b>>=1;
    }
    return ans;
}
```

需要取模:

```
int qpow(int a,int b,int mod){
    int ans=1;
    while(b){
        if(b&1) ans=ans*a%mod;
        a=a*a%mod;
        b>>=1;
    }
    return ans;
}
```