Team notebook

July 15, 2023

Contents

1 addmul

```
long long add(long long a, long long b, long long m)
{
    auto r = (a + b) % m;

    return r < 0 ? r + m : r;
}

long long mul(long long a, long long b, long long m)
{
    auto r = (a * b) % m;

    return r < 0 ? r + m : r;
}

long long fast_exp_mod(long long a, long long n, long long m) {
    long long res = 1, base = a;

    while (n) {
        if (n & 1)
            res = mul(res, base, m);
        base = mul(base, base);
        n >= 1;
    }

    return res;
}

// p is prime
long long inv(long long a, long long p) {
    return fast_exp_mod(a, p - 2, p);
}

//    assumido que (a, m) = 1
long long inverse(long long a, long long m)
{
    return fast_exp_mod(a, phi(m) - 1, m);
}

// find the inverse using extended gcd
int x, y;
int g = extended_euclidean(a, m, x, y);
if (g != 1) {
    cout << "No solution!";
}
else {
    x = (x % m + m) % m;
    cout << x << endl;
}
```

2 factorization

```cpp
#include <bits/stdc++.h>

using namespace std;

map<long long, long long> factorization(long long n) {
    map<long long, long long> fs;

    for (long long d = 2, k = 0; d * d <= n; ++d, k = 0) {
        while (n % d == 0) {
            n /= d;
            ++k;
        }

        if (k) fs[d] = k;
    }

    if (n > 1) fs[n] = 1;

    return fs;
}

map<long long, long long> factorization(long long n, vector<long long>&
    primes)
{
    map<long long, long long> fs;

    for (auto p : primes)
    {
        if (p * p > n)
            break;

        long long k = 0;

        while (n % p == 0) {
            n /= p;
            ++k;
        }

        if (k)
            fs[p] = k;
    }

    if (n > 1)
        fs[n] = 1;

    return fs;
}

int main()
{
    long long n;
    cin >> n;

    auto fs = factorization(n);
    bool first = true;

    cout << n << " = ";
    for (auto [p, k] : fs)
    {
        if (not first)
            cout << " x ";

        cout << p << "^" << k;
        first = false;
    }

    cout << endl;

    return 0;
}
```

3 gcd

```cpp
#include <bits/stdc++.h>

using namespace std;

long long gcd(long long a, long long b)
{
    return b ? gcd(b, a % b) : a;
}

long long ext_gcd(long long a, long long b, long long& x, long long& y)
{
```

```
    if (b == 0)
    {
        x = 1;
        y = 0;
        return a;
    }

    long long x1, y1;
    long long d = ext_gcd(b, a % b, x1, y1);

    x = y1;
    y = x1 - y1*(a/b);

    return d;
}

int main()
{
    long long a, b;
    cin >> a >> b;

    cout << "(" << a << ", " << b << ") = " << gcd(a, b) << '\n';

    long long x, y;
    auto d = ext_gcd(a, b, x, y);

    cout << d << " = (" << a << ")(" << x << ") + (" << b << ")(" << y <<
        ")\n";
    return 0;
}
```

4 phi

```
int phi(int n, const vector<int>& primes)
{
    if (n == 1)
        return 1;

    auto fs = factorization(n, primes);
    auto res = n;

    for (auto [p, k] : fs)
    {
        res /= p;
        res *= (p - 1);
    }

    return res;
}
```