```cpp
constexpr int N = 2e5;
constexpr int sn = 1e3;

int notprime[N], prime[N], p[N];
int split[N][3], g[N][N];
int cnt = 0;

void init_gcd() {
    notprime[1] = 1;
    int i, j, d;
    for (i = 2; i < N; i++) {
        if (!notprime[i]) {
            prime[++cnt] = i;
            p[i] = i;
        }
        for (j = 1; j <= cnt; j++) {
            if (prime[j] * i >= N) break;
            d = prime[j] * i;
            notprime[d] = 1;
            p[d] = prime[j];
            if (i % prime[j] == 0) break;
        }
    }

    split[1][0] = split[1][1] = split[1][2] = 1;
    for (i = 2; i < N; i++) {
        memcpy(split[i], split[i / p[i]], sizeof(split[i / p[i]]));
        if (split[i][0] * p[i] <= sn) split[i][0] *= p[i];
        else if (split[i][1] * p[i] <= sn) split[i][1] *= p[i];
        else split[i][2] *= p[i];
    }

    // gcd(0,0)=0 , gcd(0,x)=x
    for (i = 0; i <= sn; i++)
        for (j = 0; j <= i; j++) {
            if (!i || !j) g[i][j] = i | j;
            else g[i][j] = g[j][i] = g[j][i % j]; //j<=i
        }
}

int gcd(int x, int y) {
    int ans = 1, i, d;
    for (i = 0; i < 3; i++) {
        if (split[x][i] <= sn) d = g[split[x][i]][y % split[x][i]];
        else d = (y % split[x][i] == 0) ? split[x][i] : 1;
        ans *= d;
        y /= d; //避免算重
    }
    return ans;
}
```