GCM Solution:

Trước tiên chúng ta phân tích a, b thành các thừa số nguyên tố. Giả sử:

$$a = p_1^{k_1} \times \dots \times p_r^{k_r}; \;\; b = p_1^{l_1} \times \dots \times p_r^{l_r}$$

Chú ý rằng $(p_1, \dots p_r)$ là hợp các số nguyên tố trong cả hai phân tích của a, b. Nếu p_i không xuất hiện trong phân tích của số a thì đặt $k_i = 0$ nếu p_i không xuất hiện trong phân tích của số b thì đặt $l_i = 0$.

Để hai số x, y có $lcm(x, y) = lcm(a, b)$ và $\gcd(x, y) = \gcd(a, b)$ thì với thừa số nguyên tố p_i phải có một số có số mũ bằng k_i và số còn lại có số mũ bằng l_i.

→ Một khả năng chọn cặp x, y tương ứng với một dãy nhị phân $c_1, c_2, \dots, c_r$ trong đó $c_i = 1$ tương ứng với số mũ của p_i trong phân tích x là k_i, trong phân tích y là l_i còn $c_i = 0$ tương ứng với số mũ của p_i trong phân tích x là l_i, trong phân tích y là k_i.

Duyệt backtracking tất cả các dãy nhị phân $c_1, \dots, c_r$ với mỗi trường hợp tính x, y. Tìm hiệu $|x - y|$ nhỏ nhất.

Chú ý cách tìm tất cả các thừa số nguyên tố của cả a và b?

```
set<int> s;
s.clear();
x=a;
u=2;
while (x>1) {
    while (u≤x/u && x%u!=) ++u;
    if (u>x/u) u=x;
    s.insert(u);
    while (x%u==0) x/=u;
}
x=b;
u=2;
while (x>1) {
    while (u≤x/u && x%u!=) ++u;
    if (u>x/u) u=x;
    s.insert(u);
    while (x%u==0) x/=u;
}
r=0;
for(auto &u: s) p[++r]=u;
```