**Prime Factorisation**

**Code:**

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

ll gcd(ll a, ll b) {

    if (b == 0) return a;

    return gcd(b, a % b);

}

ll pollard\_rho(ll n) {

    ll x = 2, y = 2, d = 1;

    while (d == 1) {

        x = (x \* x + 1) % n;

        y = (y \* y + 1) % n;

        y = (y \* y + 1) % n;

        d = gcd(abs(x - y), n);

    }

    return d;

}

void factorize(ll n) {

    if (n <= 1) return;

    if (n % 2 == 0) {

        cout << 2 << " ";

        while (n % 2 == 0) n /= 2;

    }

    while (n > 1) {

        ll factor = pollard\_rho(n);

        cout << factor << " ";

        while (n % factor == 0) n /= factor;

    }

}

int main() {

    ll n;

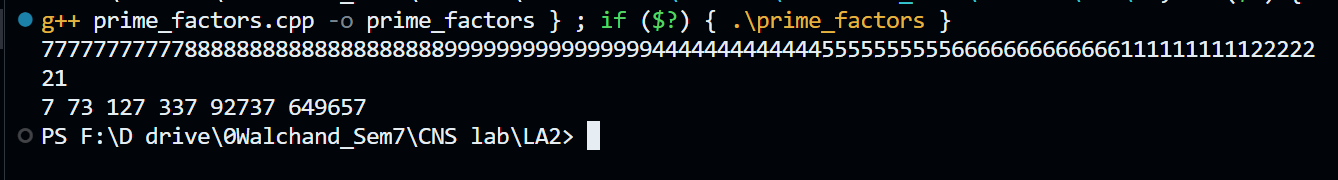
    cin >> n;

    factorize(n);

    return 0;

}

**Screenshot:**

****