Codeforces 803C (1900)

Mmukul Khedekar

27 January 2026

https://codeforces.com/problemset/problem/803/C
Accepted: https://codeforces.com/contest/803/submission/360178477

§1 Solution

§1.1 Explanation

Lemma 1.1

For an increasing sequence of length k of positive integers that sum up to n, we must have

$$\frac{k(k+1)}{2} \leq n$$

The first observation is that if d is the gcd of the k integers, then k divides n. Hence we can iterate on the divisors of n and check if there exists an increasing sequence of lenth k that sums upto n and has a gcd equal to that divisor. Using the lemma, we will have a solution if

$$\frac{k(k+1)}{2} \leq \frac{n}{d}$$

and the maximum such d will be the answer. Construction for a valid d is simple.

$$\left(d, 2d, 3d, \cdots, (k-1)d, \left(n - \tfrac{k(k-1)}{2}\right) d\right)$$

§1.2 Code

```
void solve() {
    ll n, k;
    std::cin >> n >> k;

    ll ans = -1;
    if (k > 1e6) {
        std::cout << -1 << '\n';
        return;
    }

    if ((k * (k + 1)) / 2 > n) {
        std::cout << -1 << '\n';
        return;
    }

    for (ll i = 1; i * i <= n; i++) {
        if (n % i == 0) {
            if ((k * (k + 1)) / 2 <= (n / i)) {
                ans = std::max(ans, i);
            }

            if (i * i != n) {
                if ((k * (k + 1)) / 2 <= i) {
                    ans = std::max(ans, n / i);
                }
            }
        }
    }

    if (ans == -1) {
        std::cout << -1 << ’\n’;
    } else {
        ll sum = 0;
        for (ll i = 1; i < k; i++) {
            sum += i * ans;
            std::cout << i * ans << ’ ’;
        }
        std::cout << n - sum << ’\n’;
    }
}
```