

```

struct Comb
{
    int n;
    vector<Z> _fac;
    vector<Z> _invfac;
    vector<Z> _inv;
    Comb() : n{0}, _fac{1}, _invfac{1}, _inv{0} {}
    Comb(int n) : Comb()
    {
        init(n);
    }
    void init(int m)
    {
        if (m <= n)
            return;
        _fac.resize(m + 1);
        _invfac.resize(m + 1);
        _inv.resize(m + 1);
        for (int i = n + 1; i <= m; i++)
        {
            _fac[i] = _fac[i - 1] * i;
        }
        _invfac[m] = _fac[m].inv();
        for (int i = m; i > n; i--)
        {
            _invfac[i - 1] = _invfac[i] * i;
            _inv[i] = _invfac[i] * _fac[i - 1];
        }
        n = m;
    }
    Z fac(int m)
    {
        if (m > n)
            init(2 * m);
        return _fac[m];
    }
    Z invfac(int m)
    {
        if (m > n)
            init(2 * m);
        return _invfac[m];
    }
    Z inv(int m)
    {
        if (m > n)
            init(2 * m);
        return _inv[m];
    }
    Z A(int a, int b)
    {
        if (a < b or b < 0)
            return 0;
        return fac(a) * invfac(a - b);
    }
}

```

```
z C(int n, int m)
{
    if (n < m || m < 0)
        return 0;
    return fac(n) * invfac(m) * invfac(n - m);
}
} comb;
```