```
=====================================================
======================= A ========================
#include <bits/stdc++.h>
using namespace std;
#define pb  push_back
#define mp  make_pair
#define fs  first
#define sc  second
#define ll  long long
#define all(x) x.begin(), x.end()
#define sq(x)   (x)*(x)

int a[2][2];
double eps = 1e-9;

double sqrtt(double x) {
    if (x < 1e-15) return 0;
    return sqrt(x);
}
double f(int i, int j, double x) {
    double x_0 = a[i][0];
    double res = a[i][1];
    if (j == 1) res = res - sqrtt(sq(100) - sq(x - x_0));
    if (j == 2) res = res + sqrtt(sq(100) - sq(x - x_0));
    if (j == 3) res = res + 30 - sqrtt(sq(30) - sq(x - x_0 + 40));
    if (j == 4) res = res + 30 + sqrtt(sq(30) - sq(x - x_0 + 40));
    if (j == 5) res = res + 30 - sqrtt(sq(30) - sq(x - x_0 - 40));
    if (j == 6) res = res + 30 + sqrtt(sq(30) - sq(x - x_0 - 40));
    if (j == 7) res = res - 20 - sqrtt(sq(60) - sq(x - x_0));
    if (j == 8) res = res - 20;
    return res;
}
double calc_integral(double mid, vector<vector<int> >& v) {
    vector<double> w[2];
    vector<pair<double, int> > q;
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < v[i].size(); ++j)
            w[i].pb(f(i, v[i][j], mid));
        sort(all(w[i]));
        for (int j = 0; j < w[i].size(); ++j)
            q.pb(mp(w[i][j], j % 2));
    }
    sort(all(q));
    int count = 0;
    double prev;
    double res = 0;
    for (int i = 0; i < q.size(); ++i) {
        if (count == 0) prev = q[i].fs;
        count += (q[i].sc == 0 ? 1 : -1);
        if (count == 0) res += q[i].fs - prev;
    }
    return res;
}
double integral(double be, double en, vector<vector<int> >& v) {
    double mid = (be + en) / 2.0;
    double res = calc_integral(mid, v) * (en - be);
    double ress = (calc_integral((be + mid) / 2.0, v) + calc_integral((mid + en) /
2.0, v)) * 0.5 * (en - be);
    if (abs(res - ress) < eps)
        return res;
    return integral(be, mid, v) + integral(mid, en, v);
}
double solve(int be, int en) {
    vector<vector<int> > v(2);
```

```
        for (int i = 0; i < 2; ++i) {
            int x_0 = a[i][0];
            if (-100 + x_0 <= be && be < 100 + x_0) { v[i].pb(1); v[i].pb(2); }
            if ( -70 + x_0 <= be && be < -10 + x_0) { v[i].pb(3); v[i].pb(4); }
            if (  10 + x_0 <= be && be <  70 + x_0) { v[i].pb(5); v[i].pb(6); }
            if ( -60 + x_0 <= be && be <  60 + x_0) { v[i].pb(7); v[i].pb(8); }
        }
        if (v[0].size() + v[1].size() > 0)
            return integral(be, en, v);
        return 0.0;
    }

    int main() {
        ios_base::sync_with_stdio(false);
        cin >> a[0][0] >> a[0][1] >> a[1][0] >> a[1][1];
        double res = 0;
        for (int i = -1100; i < 1100; ++i)
            res += solve(i, i + 1);
        cout << setprecision(12) << res << endl;

        return 0;
    }

=====================================================
======================= B =======================
#include <bits/stdc++.h>
using namespace std;

int const maxn = 100005;

char s[maxn];
char ans[maxn];
int d[maxn];

int main()
{
    ios_base::sync_with_stdio(false);

    cin >> s;

    int l = strlen(s);
    d[l - 1] = 1;
    for(int i = l - 2; i >= 0; --i)
        d[i] = s[i] == s[i + 1] ? d[i + 1] + 1: 1;

    int best    = 1;
    int nbest   = 0;

    for(int i = 0; i < l; ++i)
    {
        int index = i;
        for(int cur = 1; index + cur <= l; index += cur, ++cur)
            if (d[index] < cur) break;
        if (index - i > best)
        {
            best = index - i;
            nbest = i;
        }
    }

    for(int i = nbest; i < nbest + best; ++i)
        ans[i - nbest] = s[i];
    cout << ans << endl;
}
```

```
====================================================
======================= C ========================
#include <bits/stdc++.h>
using namespace std;
#define ll  long long

ll n, m;
vector<ll> p;

map<pair<ll, ll>, int> a;
// 1 - win
// 2 - lose

int solve(ll x, ll q) {
    int res = a[make_pair(x, q)];
    if (res > 0)
        return res;
    bool fl = true;
    for (int i = 0; fl && i < p.size(); ++i) {
        if (p[i] != q) {
            int y = x;
            while (fl && y % p[i] == 0) {
                y /= p[i];
                if (solve(y, p[i]) == 2)
                    fl = false;
            }
        }
    }
    a[make_pair(x, q)] = (fl ? 2 : 1);
    return (fl ? 2 : 1);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin >> n;
    m = n;
    for (ll i = 2; i * i <= m; ++i) {
        if (m % i == 0) {
            p.push_back(i);
            while (m % i == 0)
                m /= i;
        }
    }
    if (m > 1)
        p.push_back(m);

    for (int i = 0; i < p.size(); ++i)
        a[make_pair(1, p[i])] = 2;

    cout << (solve(n, 1) == 1 ? "YES" : "NO") << endl;

    return 0;
}

====================================================
======================= D ========================
#include <bits/stdc++.h>
using namespace std;
#define ll long long

const int N = 5010, M = 4, MOD = 1e9 + 7;
ll dp[M][N];
```

```
inline void clearDP(int layer) {
    for (int i = 0; i < N; ++i)
        dp[layer % M][i] = 0;
}

int main() {
    int n, k; cin >> n >> k;
    if (k > n) {
        cout << 0 <<endl;
        return 0;
    }
    dp[0][0] = 1;

    for(int i = 0; i < n; ++i) {
        for(int k = 0; k <= i; ++k) {
            dp[(i + 1) % M][k]   +=     dp[i % M][k];
            dp[(i + 2) % M][k+1] += 4 * dp[i % M][k];
            dp[(i + 3) % M][k+2] += 2 * dp[i % M][k];

            dp[(i + 1) % M][k]   %= MOD;
            dp[(i + 2) % M][k+1] %= MOD;
            dp[(i + 3) % M][k+2] %= MOD;
        }
        clearDP(i);
    }
    cout << dp[n % M][k] << endl;

    return 0;
}

=======================================================
======================== E ============================
#include <bits/stdc++.h>
using namespace std;

int ln[200005] = { };
int odd[200005] = { };
int even[200005] = { };

int main() {
    ios_base::sync_with_stdio(false);
    int n = 0, m = 0;
    cin >> n >> m;
    int a = 0, b = 0, c = 0;
    for (int i = 0; i < m; i++) {
        cin >> a >> b >> c;
        if (c) {
            even[--a]++;
            even[b]++;
        }
        else {
            odd[--a]++;
            odd[b]++;
        }
    }
    int ev_bal = 0, odd_bal = 0;
    for (int i = 0; i < n; i++) {
        ev_bal += even[i];
        if ((ev_bal % 2) && !(i % 2)) {
            ln[i] = 1;
        }
        odd_bal += odd[i];
        if ((odd_bal % 2) && (i % 2)) {
            ln[i] = 1;
            }
        }

        for (int i = 0; i < n; i++) {
            cout << ln[i] << " ";
        }
        cout << endl;

        return 0;
    }

=======================================================
======================== F ============================
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N; cin >> N;
    int ans = 0, cur = 0, b = 0;
    for (int i = 0; i < N; i++) {
        cin >> b;
        if (b) {
            ans = max(ans, cur);
            cur = 0;
        }
        else {
            cur++;
        }
    }
    ans = max(ans, cur);
    cout << ans << endl;

    return 0;
}

=======================================================
======================== G ============================
#include <bits/stdc++.h>
using namespace std;

vector<vector<pair<int, int> > > g;
vector<char> used;

void dfs(int v, int level) {
    used[v] = 1;
    for (int i = 0; i < g[v].size(); i++) {
        if (!used[g[v][i].first] && g[v][i].second >= level)
            dfs(g[v][i].first, level);
    }
}

int solve(int a, int b, int n) {
    int l = -1, r = 1e9;
    while (r - l > 1) {
        used.assign(n, 0);
        int mid = (r + l) / 2;
        dfs(a, mid);
        if (used[b])
            l = mid;
        else
            r = mid;
    }
    return l;
}
```

```
int main() {
    int n = 0, m = 0;
    scanf("%d %d", &n, &m);
    int a = 0, b = 0, c = 0;
    g.resize(n);

    for (int i = 0; i < m; i++) {
        scanf("%d %d %d", &a, &b, &c);
        g[--a].push_back(make_pair(--b, c));
        g[b].push_back(make_pair(a, c));
    }
    scanf("%d %d", &a, &b); --a; --b;
    printf("%d\n", solve(a, b, n));

    return 0;
}

=====================================================
====================== H ======================
#include <bits/stdc++.h>
using namespace std;

int const mod = 1e9 + 7;
int const maxn = 200005;

inline int safe_mul(int x, int y) {
    return x * 1LL * y % mod;
}

inline void safe_add(int& x, int y) {
    x += y;
    if (x >= mod)
        x -= mod;
}

inline int inv(int x) {
    int ans = 1;
    int b = mod - 2;
    while (b) {
        if (b & 1)
            ans = safe_mul(ans, x);
        x = safe_mul(x, x);
        b >>= 1;
    }
    return ans;
}

int fact[maxn];
int invfact[maxn];

inline int C(int n, int k) {
    return safe_mul(fact[n], safe_mul(invfact[n - k], invfact[k]));
}

void solve() {
    int n, k; cin >> n >> k;
    int t = (n - 1) / k;
    int h = t * k + 1;
    cout << C(n - h + t + 1, t + 1) << '\n';
}

int main() {
    ios_base::sync_with_stdio(false);
    fact[0] = invfact[0] = 1;
    for(int i = 1; i < maxn; ++i) {
        fact[i] = safe_mul(fact[i - 1], i);
        invfact[i] = inv(fact[i]);
    }
    int T; cin >> T;
    while (T--)
        solve();

    return 0;
}

==================================================
====================== I =========================
#include <bits/stdc++.h>
using namespace std;
#define ll long long

ll  mod = 1000000009;
ll  inv[7] = { 1, 1, 500000005, 833333341, 958333342, 591666672, 98611112 };
ll  res;
int n;
vector<ll> v, sc, ssc;
map<int, int> m;

ll C(int a, int b) {
    ll res = inv[b];
    for (int i = 0; i < b; ++i)
        res = (res * (ll)(a - i)) % mod;
    return res;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin >> n;
    for (int i = 0; i < n; ++i) {
        int a; cin >> a;
        ++m[a];
    }
    for (__typeof(m.end()) it = m.begin(); it != m.end(); ++it)
        v.push_back(it->second);
    n = v.size();

    sc.resize(n);
    sc[0] = C(v[0], 2);
    for (int i = 1; i < n; ++i)
        sc[i] = (sc[i - 1] + C(v[i], 2)) % mod;
    ssc.resize(n);
    ssc[0] = 0;
    for (int i = 1; i < n; ++i)
        ssc[i] = (ssc[i - 1] + C(v[i], 2) * sc[i - 1]) % mod;

    res = 0;
    for (int i = 0; i < n; ++i) {
        res = (res + C(v[i], 6)) % mod;
        res = (res + C(v[i], 4) * (sc[n - 1] - C(v[i], 2) + mod)) % mod;
    }
    for (int i = 2; i < n; ++i)
        res = (res + C(v[i], 2) * ssc[i - 1]) % mod;

    cout << res << endl;

    return 0;
```

```
}

================================================
====================== J =======================
#include <bits/stdc++.h>
using namespace std;

int n;
vector<int> values, sz, color, prs;
vector<vector<int> > gr, vertsWithKol;
map<vector<pair<int, int> >, int> MAP;
vector<vector<pair<int, int> > > vs;

void dfs(int v, int prv) {
    sz[v] = 1;
    prs[v] = prv;
    for (int nv : gr[v])
        if (nv != prv) {
            dfs(nv, v);
            sz[v] += sz[nv];
        }
    vertsWithKol[sz[v]].push_back(v);
}

int main() {
    ios_base::sync_with_stdio(0);
    cin >> n;
    gr.assign(n, vector<int>());
    for (int i = 1; i < n; ++i) {
        int v1, v2; cin >> v1 >> v2; --v1; --v2;
        gr[v1].push_back(v2);
        gr[v2].push_back(v1);
    }

    values.resize(n);
    for (int i = 0; i < n; ++i)
        cin >> values[i];

    sz.resize(n);
    prs.resize(n);
    vertsWithKol.resize(n+1);
    dfs(0, -1);

    vs.resize(n);
    color.assign(n, -1);

    long long ans = 0;
    for (int kol = 1; kol <= n; ++kol) {
        for (int v : vertsWithKol[kol]) {
            vs[v].push_back(make_pair(-1, values[v]));
            for (int nv : gr[v])
                if (nv != prs[v])
                    vs[v].push_back(make_pair(sz[nv], color[nv]));
            sort(vs[v].begin(), vs[v].end());
            MAP[vs[v]] = 1;
        }

        int ind = 0;
        for (auto it = MAP.begin(); it != MAP.end(); ++it)
            it->second = ind++;

        vector<int> kols(ind, 0);
        for (int v : vertsWithKol[kol]) {
            int num = MAP[vs[v]];
```

```
                ans += kols[num];
                kols[num]++;
                color[v] = num;
                vs[v].clear();
            }
            MAP.clear();
        }
        cout << ans << "\n";

        return 0;
    }
```