```cpp
#include <bits/stdc++.h>
const int INF = 1e9;
const int MOD = 1e9+7;
const long long LINF = 1e18;
#define dump(x)  cout << 'x' << ' = ' << (x) << ` `;
#define FOR(i,a,b) for(int i=(a);i<(b);++i)
#define REP(i,n) for(int i=0;i<(n);++i)
#define REPR(i,n) for(int i=n;i>=0;i--)
#define FOREACH(x,a) for(auto& (x) : (a) )
typedef long long ll;
using namespace std;
typedef pair<ll, ll> P;

// ここからライブラリ //


class weighted_union_find_tree {
private:
    vector<ll> par;
    vector<ll> ran;
    vector<ll> m_size;
    vector<ll> diff_weght;
public:
    weighted_union_find_tree(int n);
    int find(int x);
    ll weight(int x);
    ll diff(int x, int y);
    bool unite(int x, int y, int w);
    bool same(int x, int y);
    ll size(int x);
};

weighted_union_find_tree::weighted_union_find_tree(int n){
    par.resize(n);
    iota(par.begin(), par.end(), 0);
    ran.resize(n, 0);
    diff_weght.resize(n, 0);
    m_size.resize(n, 1);
};

int weighted_union_find_tree::find(int x) {
    if (par[x] == x) return x;
    else {
        int r = find(par[x]);
        diff_weght[x] += diff_weght[par[x]];
        return par[x] = r;
    }
};

ll weighted_union_find_tree::weight(int x) {
    find(x);
    return diff_weght[x];
}

ll weighted_union_find_tree::diff(int x, int y) {
    return weight(y)-weight(x);
}

bool weighted_union_find_tree::unite(int x, int y, int w) {
    w += weight(x);
    w -= weight(y);
    x = find(x);
    y = find(y);
    if (x == y) return false;
    if (ran[x] < ran[y]) swap(x,y), w *= -1;
    par[y] = x;
    m_size[x] += m_size[y];
    diff_weght[y] = w;
    if (ran[x]==ran[y]) ran[x]++;
    return true;
};

ll weighted_union_find_tree::size(int x) {
    return m_size[find(x)];
}

bool weighted_union_find_tree::same(int x, int y) {
    return (find(x) == find(y));
};
// ここまでライブラリ //

int main(int argc, char const *argv[]) {
    int n, q;
    cin >> n >> q;
    weighted_union_find_tree wuf(n);
    REP(i,q) {
        int c;
        cin >> c;
        if (!c) {
            int x,y,z;
            cin >> x >> y >> z;
            wuf.unite(x,y,z);
        }
        else {
            int x,y;
            cin >> x >> y;
            if (wuf.same(x,y)) {
                cout << wuf.diff(x,y) << endl;
            }
            else cout << "?" << endl;
        }
    }
    return 0;
}
```