#include <iostream>

using namespace std;

struct Vector3D

{

    float x, y, z;

    Vector3D()

    {

        x = 0;

        y = 0;

        z = 0;

    }

    Vector3D(float x, float y, float z)

    {

        this->x = x;

        this->y = y;

        this->z = z;

    }

    friend istream &operator>>(istream &in, Vector3D &vt)

    {

        in >> vt.x >> vt.y >> vt.z;

        return in;

    }

    friend ostream &operator<<(ostream &out, const Vector3D &vt)

    {

        out << "(" << vt.x << "," << vt.y << "," << vt.z << ")";

        return out;

    }

    friend Vector3D operator+(Vector3D v1, Vector3D v2)

    {

        float x = v1.x + v2.x;

        float y = v1.y + v2.y;

        float z = v1.z + v2.z;

        return Vector3D(x, y, z);

    }

    friend Vector3D operator-(Vector3D v1, Vector3D v2)

    {

        float x = v1.x - v2.x;

        float y = v1.y - v2.y;

        float z = v1.z - v2.z;

        return Vector3D(x, y, z);

    }

    friend Vector3D operator\*(Vector3D v1, Vector3D v2)

    {

        float x = v1.x \* v2.x;

        float y = v1.y \* v2.y;

        float z = v1.z \* v2.z;

        return Vector3D(x, y, z);

    }

};

int main()

{

    Vector3D v1, v2;

    cout << "Nhap vector v1: ";

    cin >> v1;

    cout << "Nhap vector v2: ";

    cin >> v2;

    cout << "Vector vua nhap: " << endl;

    cout << v1 << endl;

    cout << v2 << endl;

    Vector3D v3 = v1 + v2;

    cout << "v1 + v2 = " << v3 << endl;

    v3 = v1 - v2;

    cout << "v1 - v2 = " << v3 << endl;

    v3 = v1 \* v2;

    cout << "v1 \* v2 = " << v3 << endl;

}