

System:

$$k(t) = \frac{v_0 \cdot \sqrt{\alpha_1} \cdot \beta_2 \cdot \sin(\sqrt{\alpha_1 \cdot \alpha_2} \cdot t)}{\beta_1 \cdot \sqrt{\alpha_2}} + k_0 \cdot \cos(\sqrt{\alpha_1 \cdot \alpha_2} \cdot t) + \frac{\alpha_1}{\beta_1}$$

$$v(t) = v_0 \cdot \cos(\sqrt{\alpha_1 \cdot \alpha_2} \cdot t) - \frac{k_0 \cdot \sqrt{\alpha_2} \cdot \beta_1 \cdot \sin(\sqrt{\alpha_1 \cdot \alpha_2} \cdot t)}{\beta_2 \cdot \sqrt{\alpha_1}} + \frac{\alpha_2}{\beta_2}$$

```

#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>

using namespace std;

class ColumnVector {
public:
    int n;
    vector<double> v;
    ColumnVector(int m) {
        v = vector<double>(m, 0);
        n = m;
    }
    void operator=(ColumnVector a) {
        for (int i = 0; i < n; i++) {
            v[i] = a.v[i];
        }
    }
};

void plotPointsAndCurve(vector<double> ts, vector<double> vs, vector<double> ks) {
#ifdef WIN32
    FILE* pipe = _popen("c:\\gnuplot\\bin\\gnuplot -persist", "w");
#else
    FILE* pipe = popen("gnuplot -persist", "w");
#endif
    if (pipe != NULL) {
        fprintf(pipe, "%s\n", "plot '-' with lines title 'v(t)', '-' with lines title 'k(t)'");
        for (size_t i = 0; i < ts.size(); i++) {
            fprintf(pipe, "%f %f\n", ts[i], vs[i]);
        }
        fprintf(pipe, "e\n");
        for (size_t i = 0; i < ts.size(); i++) {
            fprintf(pipe, "%f %f\n", ts[i], ks[i]);
        }
        fprintf(pipe, "e\n");
        fflush(pipe);
#ifdef WIN32
        _pclose(pipe);
#else
        pclose(pipe);
#endif
    } else {
        cout << "Could not open pipe" << endl;
    }
}

void plotVvsK(vector<double> vs, vector<double> ks) {
#ifdef WIN32
    FILE* pipe = _popen("c:\\gnuplot\\bin\\gnuplot -persist", "w");
#else
    FILE* pipe = popen("gnuplot -persist", "w");
#endif
    if (pipe != NULL) {
        fprintf(pipe, "%s\n", "plot '-' with lines title 'v(k)'");
        for (size_t i = 0; i < vs.size(); i++) {
            fprintf(pipe, "%f %f\n", vs[i], ks[i]);
        }
        fprintf(pipe, "e\n");
        fflush(pipe);
#ifdef WIN32
        _pclose(pipe);
#else
        pclose(pipe);
#endif
    } else {
        cout << "Could not open pipe" << endl;
    }
}

int main() {
    double v0, k0, alpha1, beta1, alpha2, beta2;
    double T, N;

    cin >> v0 >> k0 >> alpha1 >> beta1 >> alpha2 >> beta2 >> T >> N;

    double t = T / N;

    v0 = v0 - (alpha2 / beta2);
    k0 = k0 - (alpha1 / beta1);

    vector<double> ts(N + 1, 0);
    vector<double> v(N + 1, 0);
    vector<double> k(N + 1, 0);

    // Calculate values
    for (int i = 0; i <= N; i++) {
        double j = 0 + i * t;
        ts[i] = j;
        v[i] = v0 * cos(sqrt(alpha1 * alpha2) * j) - ((k0 * sqrt(alpha2) * beta1 * sin(sqrt(alpha1 * alpha2) * j)) / (beta2 * sqrt(alpha1))) + (alpha2 / beta2);
        k[i] = (v0 * sqrt(alpha1) * beta2 * sin(sqrt(alpha1 * alpha2) * j)) / (beta1 * sqrt(alpha2)) + k0 * cos(sqrt(alpha1 * alpha2) * j) + (alpha1 / beta1);
    }

    // Plot
    plotPointsAndCurve(ts, v, k);

    // Plot v(k)
    plotVvsK(v, k);

    return 0;
}

```



