

```

// Vladislav Grigorev
// 29.04.2023
// v.grigorev@innopolis.university
/*
 * This is a C++ program that implements predator-prey model.
 * Output contains (outputting precision is 2 by default, separator is a space):
 * The array of time moments  $t_i$  in  $[0;T]$  entitled "t:"
 * The array of corresponding amount of victims  $v(t_i)$  entitled "v:"
 * The array of corresponding amount of killers  $k(t_i)$  entitled "k:"
 */
#include <iostream>
#include <vector>
#include <cmath>
#include <iomanip>
#include <cstdlib>

using namespace std;

#ifdef WIN32
#define GNUPLOT_NAME "C:\\gnuplot\\bin\\gnuplot -persist"
#endif

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

#ifdef WIN32
    FILE *pipe = _popen(GNUPLOT_NAME, "w");
#endif

    if (pipe != NULL) {
        double v0, k0, a1, b1, a2, b2, T, n;
        cin >> v0 >> k0 >> a1 >> b1 >> a2 >> b2 >> T >> n;
        vector<double> time;
        vector<double> victims;
        vector<double> killers;
        double v_new = v0 - a2 / b2;
        double k_new = k0 - a1 / b1;
        double alpha_sqrt = sqrt(a1 * a2);
        double step = T / n;
        double t = 0;
        double B = b1 / b2 * (sqrt(a2 / a1));

        // fprintf(pipe, "%s\n", "plot '-' using 1:2 title 'victims' with lines, '-' using 1:2 title 'killers' with lines");
        fprintf(pipe, "%s\n", "plot [10:55] [0:40] '-' using 1:2 title 'relation' with lines");

        for (int i = 0; i < n + 1; i++, t += step) {
            time.push_back(t);
            double v_t = v_new * cos(alpha_sqrt * t) - k_new * B * sin(alpha_sqrt * t) + a2 / b2;
            victims.push_back(v_t);
            // fprintf(pipe, "%f\t%f\n", t, v_t); // for victims graphic
        }
        // fprintf(pipe, "%s\n", "e");
        t = 0;
        for (int i = 0; i < n + 1; i++, t += step) {
            double k_t = k_new * cos(alpha_sqrt * t) + v_new / B * sin(alpha_sqrt * t) + a1 / b1;

```

```

        killers.push_back(k_t);
//        fprintf(pipe, "%f\t%f\n", t, k_t); // for killers graphic
    }
//    fprintf(pipe, "%s\n", "e");
    cout << "t:\n";
    for (int i = 0; i < n + 1; i++) {
        cout << fixed << setprecision(2) << time[i] << " ";
    }
    cout << "\nv:\n";
    for (int i = 0; i < n + 1; i++) {
        cout << fixed << setprecision(2) << victims[i] << " ";
    }
    cout << "\nk:\n";
    for (int i = 0; i < n + 1; i++) {
        cout << fixed << setprecision(2) << killers[i] << " ";
    }
    for (int i = 0; i < n + 1; i++) {
        fprintf(pipe, "%f\t%f\n", victims[i], killers[i]);
    }
    fprintf(pipe, "%s\n", "e");

} else
    cout << "Could not open pipe" << endl;

fflush(pipe);
#ifdef WIN32
    _pclose(pipe);
#endif
}

```

Graphics for: $k_0=30$; $v_0=4$; $\alpha_1=0.5471$; $\beta_1=0.0281$; $\alpha_2=0.8439$; $\beta_2=0.0266$;
 $T=30$; $n=180$.



