

CUR: Самоподобные кривые

Код

```
#include <math.h>
#include <stdio.h>

#include "labengine.h"

struct Point {
    double x;
    double y;
};

void Line(struct Point* A, struct Point* B) {
    LabDrawLine(A->x, A->y, B->x, B->y);
}

void Curve(struct Point* A, struct Point* B, int n) {
    struct Point C;
    if (n == 0) {
        C.x = B->x;
        C.y = A->y;

        Line(A, &C);
        Line(&C, B);
    }
    else {
        C.x = (A->x + B->x) / 2;
        C.y = (A->y + B->y) / 2;

        Curve(A, &C, n - 1);
        Curve(&C, B, n - 1);
    }
}

void Kokh(struct Point* A, struct Point* B, int n) {
    if (n == 0) {
        Line(A, B);
    }
    else {
        struct Point C, D, E;
        C.x = A->x + (B->x - A->x) / 3;
        C.y = A->y + (B->y - A->y) / 3;

        E.x = A->x + (B->x - A->x) * 2 / 3;
```

```

    E.y = A->y + (B->y - A->y) * 2 / 3;

    int middle_x = (C.x + E.x) / 2;
    int middle_y = (C.y + E.y) / 2;
    double dx = E.x - C.x;
    double dy = E.y - C.y;
    double length = sqrt(pow(dx, 2) + pow(dy, 2));
    double height = sqrt(3) * length / 2;
    double norm = sqrt(pow(middle_x, 2) * (1 + pow((B->x - A->x), 2) /
pow((B->y - A->y), 2))));

    struct Point NormVec;
    NormVec.x = middle_x / norm;
    NormVec.y = -middle_x * (B->x - A->x) / (norm * (B->y - A->y));

    D.x = middle_x - dy * (height / length);
    D.y = middle_y + dx * (height / length);

    Kokh(A, &C, n - 1);
    Kokh(&C, &D, n - 1);
    Kokh(&D, &E, n - 1);
    Kokh(&E, B, n - 1);
}
}

int main(void) {
    if (LabInit()) {
        struct Point A, B;
        A.x = 0;
        A.y = 500;
        B.x = 500;
        B.y = 0;

        Kokh(&A, &B, 5);
        LabDrawFlush();
        LabInputKey();
        LabTerm();
    }
    return 0;
}

```

Пример работы программы

