

```

#include<stdio.h>

double eps(double e)
{
    return 1.0 + e / 2.0 > 1.0 ? eps(e / 2.0) : e;
}

double power(double a, int b)
{
    return b == 0 ? 1 : a * power(a, b - 1);
}

double absR(double a)
{
    return a < 0 ? -a : a;
}

int main(void)
{
    double k = power(10,14);
    double a = 0.0;
    double b = 0.5;
    double x = a;
    int n = 20;

    double e = eps(1.0);
    scanf("%d%lf", &n, &k);
    printf("e = %.25lf\nn = %d\n          x          iter\n", e, n);          f(x)
    printf("taylor f(x)          taylor f(x) v.2          iter\n", e, n);

    for (int i = 0; i < n; ++i) {
        int step = 1;
        int step1 = 1;
        double lastSyepValue = 0.0;
        double currentStepValue = (1.0 + 2.0 / 3.0);
        double lastSyepValue_11 = 0.0;
        double currentStepValue_11 = (1.0);
        double lastSyepValue_12 = 0.0;
        double currentStepValue_12 = (2.0 / 3.0);
        double answer = currentStepValue;
        double answer1 = currentStepValue_11 + currentStepValue_12;
        x = a + i * (b - a) / (n - 1);

        while (absR(currentStepValue - lastSyepValue) > e * k) {
            lastSyepValue = currentStepValue;
            currentStepValue = lastSyepValue * x / 3.0 + (2.0 / 3.0) * power(x, step);
            answer += currentStepValue;
            step++;
        }
        while (absR((currentStepValue_11 + currentStepValue_12) - (lastSyepValue_11 +
lastSyepValue_12))) > e * k) {
            lastSyepValue_11 = currentStepValue_11;
            lastSyepValue_12 = currentStepValue_12;
            currentStepValue_11 = lastSyepValue_11 * x;
            currentStepValue_12 = lastSyepValue_12 * x / 3.0;
            answer1 += currentStepValue_11 + currentStepValue_12;
            step1++;
        }

        printf("%.25lf %.25lf %.25lf %.25lf  %d  %d\n", x, (3.0 * x - 5.0) / (x
* x - 4.0 * x + 3.0), -answer, -answer1, step, step1);
    }
    return 0;
}

```