

```

#include<iostream>
#include<vector>
#include<assert.h>
#include<algorithm>
#include<string>
#include <windows.h>
using namespace std;

double f(double x) // Подсчёт значения y в точке x
{
    return x * x - 2 * cos(x) + 1;
}
double X(double x)
{
    return sqrt(2 * cos(x) - 1);
}

int main()
{
    double a, b, E, x, count = 1, Z, A, B;
    SetConsoleOutputCP(CP_UTF8);
    cout << "Введите концы отрезка a и b, а так же точность E: ";
    cin >> a >> b >> E;
    A = a;
    B = b;
    x = (a + b) / 2;
    cout << "\nМетод дихотомии:\n";
    while (!(f(x+E) > 0 && f(x - E) < 0) || (f(x + E) < 0 && f(x - E) > 0))
    {
        if (f(x) < 0)
        {
            if (f(a) > 0)
                b = a;
            else
                b = b;
        }
        else
        {
            if (f(a) < 0)
                b = a;
            else
                b = b;
        }
        a = x;
        x = (a + b) / 2;
        count++;
    }
    cout << x << " " << E << " " << count;

    count = 1;
    double fpXa = (1.0 / 2.0) / (sqrt(X(a))); // Производная от выраженного x = F(x)
    double fpXb = (1.0 / 2.0) / (sqrt(X(b)));
    cout << "\n\nМетод простых итераций:";

    if (fpXa < 1)
    {
        Z = X(A);
        x = X(Z);
        /*while (abs(X(Z) - X(x)) > E)*/
        while (abs(x - Z) > E)
        {
            x = X(x);
            Z = X(Z);
            count++;
        }
    }
}

```

```

        cout << endl << x << " " << E << " " << count;
    }
    else if (fpXb < 1)
    {
        Z = X(B);
        x = X(Z);
        /*while (abs(X(Z) - X(x)) > E)*/
        while (abs(x - Z) > E || f(x) == 0)
        {
            x = X(x);
            Z = X(Z);
            count++;
        }
        cout << endl << x << " " << E << " " << count;
    }
    else
        cout << "\nМетод простых итераций неприменим с данными значениями.";
}

```