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

const int POINT_NUM = 25;     // データ点数（等間隔点またはチェビシェフ点）
const int INTER_NUM = 101;    // 補間結果を求める際に使用する点数（描画用）
const double PI = 3.1415;  // 円周率

// ----- 構造体定義 -----

// データ点用の構造体
typedef struct {
    double x[POINT_NUM];  // x座標
    double y[POINT_NUM];  // 対応する関数値 y = f(x)
} PointSet;

// 補間評価用の構造体
typedef struct {
    double x[INTER_NUM];       // 評価点
    double f[INTER_NUM];       // 元の関数値 f(x)
    double lag_eq[INTER_NUM];  // 等間隔点を使ったラグランジュ補間結果
    double lag_ch[INTER_NUM];  // チェビシェフ点を使ったラグランジュ補間結果
} EvalResult;

// ----- 関数定義 -----

// 元の関数 f(x)
double f(double x) {
    return 1.0 / (4.0 + 100.0 * x);
}

// ラグランジュ補間の計算
double lagrange(double x[], double y[], int n, double xp) {
    double result = 0.0;
    for (int i = 0; i < n; i++) {
        double term = y[i];
        for (int j = 0; j < n; j++) {
            if (j != i) {
                term *= (xp - x[j]) / (x[i] - x[j]);
            }
        }
        result += term;
    }
    return result;
}

// 等間隔点を生成し、その関数値も同時に計算
void generate_equally_spaced(PointSet *p, double xmin, double xmax, int n) {
    for (int i = 0; i < n; i++) {
        (*p).x[i] = xmin + (xmax - xmin) * i / (n - 1);
        (*p).y[i] = f((*p).x[i]);
    }
}

// チェビシェフ点を生成し、その関数値も同時に計算
void generate_chebyshev_nodes(PointSet *p, double xmin, double xmax, int n) {
    for (int i = 0; i < n; i++) {
        (*p).x[i] = 0.5 * (xmin + xmax) +
                    0.5 * (xmax - xmin) * cos((2.0 * i + 1.0) * PI / (2.0 * n));
        (*p).y[i] = f((*p).x[i]);
    }
}

// 補間を評価する点とその関数値を生成
void generate_eval_points(EvalResult *r, double xmin, double xmax, int n) {
    for (int i = 0; i < n; i++) {
        (*r).x[i] = xmin + (xmax - xmin) * i / (n - 1);
        (*r).f[i] = f((*r).x[i]);
    }
}

// CSVファイルに結果を出力
void write_csv(const char *filename,const PointSet *eq, const PointSet *ch, const EvalResult *r) {

    FILE *fp = fopen(filename, "w");
    if (!fp) {
```

```
        printf("Could not open output file: %s\n", filename);
        return;
    }

    fprintf(fp, "x_eq,y_eq,x_ch,y_ch,x_eval,L_eq(x),L_ch(x)\n");
    for (int i = 0; i < INTER_NUM; i++) {
        if (i < POINT_NUM)
            fprintf(fp, "%.8f,%.8f,%.8f,%.8f,",
                    (*eq).x[i], (*eq).y[i], (*ch).x[i], (*ch).y[i]);
        else
            fprintf(fp, ",,,,");

        fprintf(fp, "%.8f,%.8f,%.8f\n",
                (*r).x[i], (*r).lag_eq[i], (*r).lag_ch[i]);
    }

    fclose(fp);
    printf("Interpolation results saved to %s\n", filename);
}

// ----- メイン関数 -----

int main(int argc, char *argv[]) {
    if (argc != 2) { // 実行時コマンドライン引数の数が2個でなかったらエラーとする
        fprintf(stderr, "Usage: %s outputfile\n", argv[0]);
        return 1;
    }

    const char *filename = argv[1];

    double xmin = -1.0, xmax = 1.0;

    // 構造体の宣言と初期化
    PointSet eq = {}, ch = {};
    EvalResult eval = {};

    // 等間隔点とチェビシェフ点の生成
    generate_equally_spaced(&eq, xmin, xmax, POINT_NUM);
    generate_chebyshev_nodes(&ch, xmin, xmax, POINT_NUM);

    // 補間の評価点を生成
    generate_eval_points(&eval, xmin, xmax, INTER_NUM);

    // 補間結果を評価点上で計算
    for (int i = 0; i < INTER_NUM; i++) {
        eval.lag_eq[i] = lagrange(eq.x, eq.y, POINT_NUM, eval.x[i]);
        eval.lag_ch[i] = lagrange(ch.x, ch.y, POINT_NUM, eval.x[i]);
    }

    // 結果をCSVに出力
    write_csv(filename, &eq, &ch, &eval);

    return 0;
}
```