```c
/* slowFT-linear-filter.c, by S. Tanaka, 2006 */
#include <math.h>
#include <stdio.h>
#include <time.h>

#define twoPi 3.1415926535 * 2.0
#define N     512

FILE *fpR, *fpW;

void main( void )
{
    int    t, fn, fdmy[N];
    double dataR[N], DataR[N], DataI[N], iDataR[N], iDataI[N];
    static double wR[N][N], wI[N][N];
    double ifR, ifI, pdmy;
    char   ldmy0[100],ldmy1[20],ldmy2[20],ldmy3[20],ldmy4[20],ldmy5[20];

    fpR = fopen("slowFT-linear.txt", "r");

    fseek(fpR, 0L, SEEK_SET);
    fscanf(fpR, " %s ", &ldmy0);
    fscanf(fpR, " %s %s %s %s %s", &ldmy1, &ldmy2, &ldmy3, &ldmy4, &ldmy5);
    for (fn=0; fn<N; fn++) {
        fscanf(fpR," %d %lf %lf %lf %lf ",
            &fdmy[fn], &dataR[fn], &DataR[fn], &DataI[fn], &pdmy);
        printf(" %3d %10.6f %10.6f %10.6f %10.6f ¥n",
            fdmy[fn], dataR[fn], DataR[fn], DataI[fn], pdmy);
    }
    for (fn=0; fn<N; fn++) {
        for (t=0; t<N; t++){
            wR[t][fn] =  cos( -fn*twoPi*t/N );
            wI[t][fn] = -sin( -fn*twoPi*t/N );
        }
    }

    //特定周波数を取り除く（零にする）
    DataR[  2] = 0.0; DataI[  2] = 0.0;
    DataR[510] = 0.0; DataI[510] = 0.0;
    ifR = 0.0;
    ifI = 0.0;
    for (t=0; t<N; t++) {
        for (fn=0; fn<N; fn++) {
            ifR      = ifR + DataR[fn]*wR[t][fn] - DataI[fn]*wI[t][fn];
            ifI      = ifI + DataR[fn]*wI[t][fn] + DataI[fn]*wR[t][fn];
        }
        iDataR[t] = ifR/N;
        iDataI[t] = ifI/N;
        ifR = 0.0;
        ifI = 0.0;
    }

    fpW = fopen("slowInvFT-linear-filter.txt", "w");
    fprintf(fpW, "              離散 逆フーリエ変換¥n");
    fprintf(fpW, "t,fn   Re(F(fn))   Im(F(fn))   逆変換後実数   逆変換後虚数¥n");

    for (t=0; t<N; t++) {
        fprintf(fpW," %3d %10.6f %10.6f %10.6f %10.6f ¥n",
            t, DataR[t], DataI[t], iDataR[t], iDataI[t]);
    }
    printf("Result written to 'slowInvFT-linear-filter.txt.'¥n");

}
```