```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <math.h>
#include <string.h>

struct dpoint{
    double key;
    int n;
    double f;
};

//generate mas from file
int
generateMas(struct dpoint* mas, char* name)
{
    int n=0;
    double tmp = 0.0;
    FILE* in = fopen(name,"r");
    fscanf(in,"%lf",&tmp);
    mas[0].key = tmp;
    mas[0].n = 1;
    n = 1;
    while(fscanf(in,"%lf",&tmp) != EOF){
        int flg = 1;
        for(int i = 0; i < n; i++){
            if(tmp == mas[i].key){
                mas[i].n++;
                flg = 0;
                break;
            }
        }
        if(flg){
            mas[n].key = tmp;
            mas[n].n = 1;
            n++;
        }
    }
    fclose(in);
    int sum = 0;
    for (int i = 0; i < n; ++i){
        sum += mas[i].n;
    }
    for (int i = 0; i < n; ++i){
        mas[i].f = (double)mas[i].n/sum;
    }
    return n;
}

// swap mas[a] and mas[b]
void
swap(struct dpoint* mas, int a, int b)
{
    struct dpoint tmp = mas[a];
    mas[a] = mas[b];
    mas[b] = tmp;
}

```

```
//sort mas
void
sort(struct dpoint* mas, int n)
{
    for (int i = 0; i < n-1; i++){
        for (int j = i+1; j < n; j++){
            if (mas[j].key < mas[i].key){
                swap(mas,i,j);
            }
        }
    }
}

//generate Var-row
void
generateVar(struct dpoint* mas, int n)
{
    FILE* out = fopen("V.txt","w");
    for (int i = 0; i < n; i++){
        fprintf(out, "|%.3lf | %ld |\n",mas[i].key,mas[i].n);
        fprintf(out, "__________________\n");
    }
    fclose(out);
}

void
genDataABS(struct dpoint* mas, int n)
{
    FILE* out = fopen("DPh.txt","w");
    for (int i = 0; i < n; ++i){
        fprintf(out, "%d %.3lf\n",mas[i].n,mas[i].key);
    }
    fclose(out);
}

void
genDataRel(struct dpoint* mas, int n)
{
    FILE* out = fopen("DPh1.txt","w");
    for (int i = 0; i < n; ++i){
        fprintf(out, "%.3lf %.3lf\n",mas[i].f,mas[i].key);
    }
    fclose(out);
}

void
genDataF(struct dpoint* mas, int n)
{
    FILE* out = fopen("F.txt","w");
    double sum = 0;
    fprintf(out, "0.000 %lf\n",-2*abs(mas[0].key));
    for (int i = 0; i < n; ++i)
    {
        fprintf(out, "%.3lf %.3lf\n\n",sum,mas[i].key);
        sum+= mas[i].f;
        fprintf(out, "%.3lf %.3lf\n",sum,mas[i].key);
    }
    fprintf(out, "1.000 %.3lf\n",2*mas[n-1].key);
    fclose(out);
}

void
genDigParam(struct dpoint* mas, int n)
{
    FILE* out = fopen("Dig.txt","w");
    double x = 0.0;
    double D = 0.0;
    double sig = 0.0;
    double S = 0.0;
    int sum = 0;
    for (int i = 0; i < n; ++i){
        x+= mas[i].key*mas[i].n;
        D+= mas[i].key*mas[i].n*mas[i].key;
        sum += mas[i].n;
    }
    x /= sum;
    D /= sum;
    D -= x*x;
    sig = sqrt(D);
    S = D*(sum-1)/sum;
    fprintf(out, "X выборочное = %- -.3lf\n",x);
    fprintf(out, "D выб = %- -.3lf\n",D);
    fprintf(out, "среднее кв.отклонение = %- -.3lf\n",sig);
    fprintf(out, "S = %- -.3lf\n",S);
    fclose(out);
}

//удаляем данные
void
delData(int tmp)
{
    switch(tmp){
        case 1:
            system("rm V.txt");
            break;
        case 2:
            system("rm RelPh.png");
            system("rm AbsPh.png");
            system("rm DPh.txt");
            system("rm DPh1.txt");
            break;
        case 3:
            system("rm F.txt");
            system("rm F.png");
            break;
        case 4:
            system("rm Dig.txt");
            break;
        default:
            break;
    }
}

int
main(void)
{
    struct dpoint mas[100];
    int n = 0;
    n = generateMas(mas,"data.txt");
    sort(mas,n);
    printf("1 - составление вариационного ряда\n");
    printf("2 - составления ряда частот и относительных частот\n");
    printf("3 - постоение F(x) \n");
    printf("4 - вычисление характеристик(Xв,Dв,sig,S\n");
    int tmp;
    scanf("%d",&tmp);
    printf("N is %d\n",n);
    switch(tmp){
        case 1:
            generateVar(mas,n);
            system("subl V.txt");
            break;
        case 2:
            genDataABS(mas,n);
            genDataRel(mas,n);
            system("gnuplot scr_2_1.txt");
            system("gnuplot scr_2_2.txt");
            system("ristretto AbsPh.png");
            break;
        case 3:
            genDataF(mas,n);
            system("gnuplot scr_3.txt");
            system("ristretto F.png");
            break;
        case 4:
            genDigParam(mas,n);
            system("subl Dig.txt");
            break;
        default:
            break;
    }
    printf("Введите целое число для завершения\n");
    scanf("%d",&n);
    delData(tmp);
    return 0;
}
```