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

struct dpoint{
    double a;
    double b;
    double cent;
    int n;
    double f;
};

int
generateMas(struct dpoint* mas, char* name)
{
    int n=0;
    FILE* in = fopen(name,"r");
    n = 0;
    while(fscanf(in,"%lf %lf %d",&mas[n].a,&mas[n].b,&mas[n].n) != EOF){
        n++;
    }
    fclose(in);
    return n;
}

void
HandleInput(struct dpoint* mas, int n)
{
    for (int i = 0; i < n; i++){
        printf("(a,b),n [%d]:\n",i);
        scanf("%lf %lf %d",&mas[i].a,&mas[i].b,&mas[i].n);
    }
}

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

void
genGistDataRel(struct dpoint* mas, int n)
{
    FILE* out = fopen("Rel.txt","w");
    for (int i = 0; i < n; i++){
            fprintf(out, "0 %.3lf\n",mas[i].a);
            fprintf(out, "%.3lf %.3lf\n",mas[i].f,mas[i].a);
            fprintf(out, "%.3lf %.3lf\n",mas[i].f,mas[i].b);
            fprintf(out, "0 %.3lf\n",mas[i].b);
```

```
        }
    fclose(out);
}

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

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

void
genFDataInt(struct dpoint* mas, int n)
{
    FILE* out = fopen("FInt.txt","w");
    double sum = 0.0;
    fprintf(out, "0 %lf\n",-2*mas[0].a);
    fprintf(out, "0 %lf\n\n",mas[0].a);
    for (int i = 0; i < n; ++i){
        sum += mas[i].f;
        fprintf(out, "%.3lf %.3lf\n",sum,mas[i].a);
        fprintf(out, "%.3lf %.3lf\n\n",sum,mas[i].b);
    }
    fprintf(out, "1.000 %.3lf\n",mas[n-1].b);
    fprintf(out, "1.000 %.3lf\n",2*mas[n-1].b);
    fclose(out);
}

void
genFDataGroup(struct dpoint* mas, int n)
{
    FILE* out = fopen("FGroup.txt","w");
    double sum = 0.0;
    fprintf(out, "0 %lf\n",-2*mas[0].cent);
    fprintf(out, "0 %lf\n\n",mas[0].cent);
    for (int i = 0; i < n-1; ++i){
        sum += mas[i].f;
        fprintf(out, "%.3lf %.3lf\n",sum,mas[i].cent);
        fprintf(out, "%.3lf %.3lf\n\n",sum,mas[i+1].cent);
    }
    fprintf(out, "1.000 %.3lf\n",mas[n-1].cent);
    fprintf(out, "1.000 %.3lf\n",2*mas[n-1].cent);
    fclose(out);
}

```

```
117. void
118. genDigParam(struct dpoint* mas, int n)
119. {
120.     FILE* out = fopen("Dig.txt","w");
121.     double x = 0.0;
122.     double D = 0.0;
123.     double sig = 0.0;
124.     double S = 0.0;
125.     int sum = 0;
126.     for (int i = 0; i < n; ++i){
127.         x+= mas[i].cent*mas[i].n;
128.         D+= mas[i].cent*mas[i].n*mas[i].cent;
129.         sum += mas[i].n;
130.     }
131.     x /= sum;
132.     D /= sum;
133.     D -= x*x;
134.     sig = sqrt(D);
135.     S = D*(sum-1)/sum;
136.     fprintf(out, "X выборочное = %- -.6lf\n",x);
137.     fprintf(out, "D выб = %- -.6lf\n",D);
138.     fprintf(out, "среднее кв.отклонение = %- -.6lf\n",sig);
139.     fprintf(out, "S = %- -.6lf\n",S);
140.     fclose(out);
141. }
142. 
143. void
144. delData(int tmp)
145. {
146.     switch(tmp){
147.         case 1:
148.             system("rm Abs.png");
149.             system("rm Rel.png");
150.             system("rm Abs.txt");
151.             system("rm Rel.txt");
152.             break;
153.         case 2:
154.             system("rm Abs.png");
155.             system("rm Rel.png");
156.             system("rm Abs.txt");
157.             system("rm Rel.txt");
158.             break;
159.         case 3:
160.             system("rm FGroup.png");
161.             system("rm FGroup.txt");
162.             system("rm FInt.png");
163.             system("rm FInt.txt");
164.             break;
165.         case 4:
166.             system("rm Dig.txt");
167.             break;
168.         default:
169.             break;
170.     }
171. }
172. 
173. int
174. main(void)
```

```
{
    struct dpoint mas[100];
    int n = 0;
    int tmp;
    printf("0 - ввод данных их файла\n");
    printf("1 - ручной ввод\n");
    scanf("%d",&tmp);
    if(tmp){
        printf("Число интервалов n\n");
        scanf("%d",&n);
        HandleInput(mas,n);
    }else{
        n = generateMas(mas,"data.txt");
    }
    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;
        mas[i].cent = (mas[i].a + mas[i].b)/2;
    }
    printf("1 - интервальный ряд частот и относительных частот(гистограммы)\n");
    printf("2 - группированный ряд частот и относительных частот(полигоны)\n");
    printf("3 - постоение F(x) \n");
    printf("4 - вычисление характеристик(Хв,Dв,sig,S\n");
    scanf("%d",&tmp);
    switch(tmp){
        case 1:
            genGistDataAbs(mas,n);
            genGistDataRel(mas,n);
            system("gnuplot scr_1_1.txt");
            system("gnuplot scr_1_2.txt");
            system("ristretto Abs.png");
            break;
        case 2:
            genPolDataAbs(mas,n);
            genPolDataRel(mas,n);
            system("gnuplot scr_1_1.txt");
            system("gnuplot scr_1_2.txt");
            system("ristretto Abs.png");
            break;
        case 3:
            genFDataInt(mas,n);
            genFDataGroup(mas,n);
            system("gnuplot scr_3_1.txt");
            system("gnuplot scr_3_2.txt");
            system("ristretto FInt.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;
}
```