```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;
    double pbin;
};

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);
}

int
generateMas(struct dpoint* mas, char* name, int flg)
{
    int n=0;
    FILE* in = fopen(name,"r");
    n = 0;
    double tmp;
    if(flg){
        int sum = 0;
        while(fscanf(in,"%lf %lf %d",&mas[n].a,&mas[n].b,&mas[n].n) != EOF){
        n++;
        }
        for(int i = 0; i < n; i++){
            mas[i].cent = (mas[i].a + mas[i].b)/2;
            sum += mas[i].n;
        }
        for(int i = 0; i < n; i++){
            mas[i].f = (double)mas[i].n/sum;
        }
    }else{
        fscanf(in,"%lf",&tmp);
        mas[0].cent = 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].cent){
                    mas[i].n++;
                    flg = 0;
                    break;
                }
            }
            if(flg){
```

```
                mas[n].cent = tmp;
                mas[n].n = 1;
                n++;
            }
        }
        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;
        }
    }
    fclose(in);
    return n;
}

int
HandleInput(struct dpoint* mas, int n, int flg)
{
    int sum = 0;
    int size = 1;
    if(flg){
        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);
        }
        for(int i = 0; i < n; i++){
            mas[i].cent = (mas[i].a + mas[i].b)/2;
            sum += mas[i].n;
        }
        for(int i = 0; i < n; i++){
            mas[i].f = (double)mas[i].n/sum;
        }
    }else{
        printf("Mas of data is:\n");
        double tmp;
        scanf("%lf",&mas[0].cent);
        mas[0].n = 1;
        for(int i = 1; i < n; i++){
            scanf("%lf",&tmp);
            int flg1 = 1;
            for(int j = 0; j < size; j++){
                if(tmp == mas[j].cent){
                    mas[j].n++;
                    flg1 = 0;
                    break;
                }
            }
            if(flg1){
                mas[size].cent = tmp;
                mas[size].n = 1;
                size++;
            }
        }
        for(int i = 0; i < size; i++){
            sum += mas[i].n;
        }
        for(int i = 0; i < size; i++){
            mas[i].f = (double)mas[i].n/sum;
        }
    }
```

```
125.     return (flg)?n:size;
126. }
127. 
128. double
129. F0(double a)
130. {
131.     double res = erf(a/sqrt(2))/2;
132.     return res;
133. }
134. 
135. int
136. isA(double a)
137. {
138.     if(a == 0.995)
139.         return 0;
140.     if(a == 0.99)
141.         return 1;
142.     if(a == 0.975)
143.         return 2;
144.     if(a == 0.95)
145.         return 3;
146.     if(a == 0.9)
147.         return 4;
148.     if(a == 0.75)
149.         return 5;
150.     if(a == 0.5)
151.         return 6;
152.     if(a == 0.25)
153.         return 7;
154.     if(a == 0.1)
155.         return 8;
156.     if(a == 0.05)
157.         return 9;
158.     if(a == 0.025)
159.         return 10;
160.     if(a == 0.01)
161.         return 11;
162.     if(a == 0.005)
163.         return 12;
164.     return -1;
165. }
166. 
167. double
168. GetXiSq(double a, int k)
169. {
170.     FILE* in = fopen("xi2Table.txt","r");
171.     int key = 0;
172.     double xi[13];
173.     while(key != k){
174.         fscanf(in,"%d %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf
%lf",&key,&xi[0],&xi[1],&xi[2],&xi[3],&xi[4],&xi[5],&xi[6],&xi[7],&xi[8],&xi[9],&xi[10],&xi[11],&xi[12]);
175.     }
176.     int num = isA(a);
177.     fclose(in);
178.     return (num == -1)?-1:xi[num];
179. }
180. 
181. void
182. genPolDataAbs(struct dpoint* mas, int n)
183. {
184.     FILE* out = fopen("Abskr.txt","w");
185.     for (int i = 0; i < n; ++i){
```

```
        fprintf(out, "%d %.3lf\n",mas[i].n,mas[i].cent);
    }
    fclose(out);
}

void
genPolDataAbs1(struct dpoint* mas, int n)
{
    FILE* out = fopen("Abskr.txt","w");
    for (int i = 0; i < n; ++i){
        fprintf(out, "%.3lf %.3lf\n",mas[i].pbin,mas[i].cent);
    }
    fclose(out);
}

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

void
genScr(double a, double b, int y, int tmp2)
{
    FILE* scr = fopen("scr_1.txt","w");
    fprintf(scr, "set terminal png\n");
    fprintf(scr, "set output 'Abs.png'\n");
    fprintf(scr, "set xrange [%lf:%lf]\n",a,b);
    fprintf(scr, "set yrange [0:%ld]\n",y);
    fprintf(scr, "set multiplot\n");
    if(tmp2)
        fprintf(scr, "plot \"Abs.txt\" u 2:1 w l lw 4\n");
    fprintf(scr, "plot \"Abskr.txt\" u 2:1 w l lw 4 lt rgb 'red'\n");
    fclose(scr);
}

void
genScrN(double a, double sig, double x, double y)
{
    FILE* scr = fopen("scr_2.txt","w");
    fprintf(scr, "set terminal png\n");
    fprintf(scr, "set output 'Abs1.png'\n");
    fprintf(scr, "set xrange [%lf:%lf]\n",x,y);
    fprintf(scr, "plot 1/(sqrt(2*pi)*%lf)*exp(-(x-%lf)**2/(2*%lf**2)) lw 4 lt rgb 'red'",sig,a,sig);
    fprintf(scr, "plot \"Abskr.txt\" u 2:1 w l lw 4 lt rgb 'red'\n");
    fclose(scr);
}


void
genScrBin(double a, double b)
{
    FILE* scr = fopen("scr_bin.txt","w");
    fprintf(scr, "set terminal png\n");
    fprintf(scr, "set output 'TeorBin.png'\n");
    fprintf(scr, "set xrange [%lf:%lf]\n",a,b);
    // fprintf(scr, "set yrange [0:1]\n");
    fprintf(scr, "plot \"Abskr.txt\" u 2:1 w l lw 4 lt rgb 'red'\n");
```

```
248.     fclose(scr);
249. }
250. 
251. void
252. genScrBinT(double a, double b)
253. {
254.     FILE* scr = fopen("scr_bint.txt","w");
255.     fprintf(scr, "set terminal png\n");
256.     fprintf(scr, "set output 'bin.png'\n");
257.     fprintf(scr, "set xrange [%lf:%lf]\n",a,b);
258.     // fprintf(scr, "set yrange [0:1]\n");
259.     fprintf(scr, "plot \"Absrel.txt\" u 2:1 w l lw 4 lt rgb 'red'\n");
260.     fclose(scr);
261. }
262. 
263. void
264. swap(struct dpoint* mas, int i, int j)
265. {
266.     struct dpoint tmp = mas[i];
267.     mas[i] = mas[j];
268.     mas[j] = tmp;
269. }
270. 
271. void
272. sort(struct dpoint* mas, int n)
273. {
274.     for (int i = 0; i < n-1; i++){
275.         for (int j = i+1; j < n; j++){
276.             if (mas[j].cent < mas[i].cent){
277.                 swap(mas,i,j);
278.             }
279.         }
280.     }
281. }
282. 
283. void
284. genPiData(struct dpoint* mas, int n, double* p, int tmp2)
285. {
286.     FILE* out = fopen("Pi.txt","w");
287.     if(tmp2){
288.         for(int i = 0; i < n; i++){
289.             fprintf(out, "(%.3lf, %.3lf) | %.3lf\n",mas[i].a,mas[i].b,p[i]);
290.         }
291.     }else{
292.         for(int i = 0; i < n; i++){
293.             fprintf(out, "%.3lf | %.3lf\n",mas[i].cent,p[i]);
294.         }
295.     }
296.     fclose(out);
297.     system("subl Pi.txt");
298. }
299. 
300. void
301. CheckNorm(struct dpoint* mas, int n, double a, int tmp2)
302. {
303.     double p[n];
304.     double np[n];
305.     double row[n];
306.     double x = 0.0;
307.     double D = 0.0;
308.     double sig = 0.0;
309.     int sum = 0;
```

```
    int mx = 0;
    for(int i = 0; i < n; i++){
        x += mas[i].cent * mas[i].n;
        D += mas[i].cent * mas[i].n * mas[i].cent;
        sum += mas[i].n;
        if(mas[i].n > mx){
            mx = mas[i].n;
        }
    }
    x /= sum;
    D /= sum;
    D -= x*x;
    sig = sqrt(D);
    printf("Хв = sum(XiNi)/N,(i=1,%ld) = %.3lf\n",n,x);
    printf("Dв = sum(Xi^2Ni)/N - Хв^2,(i=1,%ld) = %.3lf\n",n,D);
    printf("sig = sqrt(Dв) = %.3lf\n\n",sig);
    printf("Предположительное мат ожидание = %.3lf\n",x);
    printf("Предположительное ср.кв отклонение = %.3lf\n",sig);
    if(tmp2){
        p[0] = F0((mas[0].b-x)/sig) + 0.5;
        for(int i = 1; i < n-1; i++){
            p[i] = F0((mas[i].b-x)/sig) - F0((mas[i].a-x)/sig);
        }
        p[n-1] = 0.5 - F0((mas[n-1].a-x)/sig);
    }else{
        p[0] = F0((mas[1].cent - x)/sig) + 0.5;
        for(int i = 1; i < n-1; i++){
            p[i] = F0((mas[i+1].cent - x)/sig) - F0((mas[i].cent - x)/sig);
        }
        p[n-1] = 0.5 - F0((mas[n-1].cent - x)/sig);
    }
    double X2 = 0;
    for(int i = 0; i < n; i++){
        np[i] = p[i]*sum;
        row[i] = (mas[i].n-np[i])*(mas[i].n-np[i])/np[i];
        X2 += row[i];
    }
    printf("X2: %.5lf\n",X2);
    double sr = GetXiSq(a,n-3);
    printf("xi(table) %.5lf\n",sr);
    if(tmp2)
        genGistDataAbs(mas,n);
    genPolDataAbs(mas,n);
    if(tmp2)
        genScr(mas[0].a,mas[n-1].b,mx,tmp2);
    else
        genScr(mas[0].cent,mas[n-1].cent,mx,tmp2);
    if(tmp2)
        genScrN(x,sig,mas[0].a,mas[n-1].b);
    else
        genScrN(x,sig,mas[0].cent,mas[n-1].cent);
    system("gnuplot scr_1.txt");
    system("gnuplot scr_2.txt");
    genPiData(mas,n,p,tmp2);
    if(sr > 0){
        if(X2 <= sr){
            printf("X2 <= XiTable\n");
            printf("Гипотеза подтверждена\n");
        }else{
            printf("X2 > XiTable\n");
            printf("Гипотеза опровергнута\n");
        }
```

```
    }else{
        printf("a не найдена в таблице или количество интервалов/точек меньше 4\n");
    }
    system("ristretto Abs.png");
}

void
CheckBin(struct dpoint* mas, int n, double a, double cnt, int tmp2)
{
    double x = 0.0;
    double D = 0.0;
    double sig = 0.0;
    double np[n];
    double row[n];
    double X2 = 0;
    int sum = 0;
    int mx = 0;
    for(int i = 0; i < n; i++){
        x += mas[i].cent * mas[i].n;
        D += mas[i].cent * mas[i].n * mas[i].cent;
        sum += mas[i].n;
        if(mas[i].n > mx){
            mx = mas[i].n;
        }
    }
    x /= sum;
    D /= sum;
    D -= x*x;
    sig = sqrt(D);
    double p = x/cnt;
    printf("p = Хв/N, N - число испытаний\n");
    printf("Хв = sum(XiNi)/N,(i=1,%ld) = %.3lf\n",n,x);
    printf("Dв = sum(Xi^2Ni)/N - Хв^2,(i=1,%ld) = %.3lf\n",n,D);
    printf("sig = sqrt(Dв) = %.3lf\n\n",sig);
    printf("Pi = C_from_n_by_i * p^i * q^(N-i)\n");
    double q = -p + 1;
    for(int i = 0; i < n; i++){
        mas[i].pbin = tgamma(cnt+1)/(tgamma(cnt-i+1)*tgamma(i+1))*pow(p,i)*pow(q,cnt-i);
    }
    // reconstr(mas,n,start,finish);
    for(int i = 0; i < n; i++){
        np[i] = mas[i].pbin*sum;
        row[i] = (mas[i].n-np[i])*(mas[i].n-np[i])/np[i];
        X2 += row[i];
        printf("%.3lf\n",mas[i].pbin);
    }
    genPolDataAbs1(mas,n);
    genPolDataRel(mas,n);
    genScrBin(mas[0].cent,mas[n-1].cent);
    genScrBinT(mas[0].cent,mas[n-1].cent);
    system("gnuplot scr_bin.txt");
    system("gnuplot scr_bint.txt");
    system("ristretto bin.png");
    printf("X2: %.5lf\n",X2);
    double sr = GetXiSq(a,n-2);
    printf("xi(table) %.5lf\n",sr);
    if(sr > 0){
        if(X2 <= sr){
            printf("X2 <= XiTable\n");
            printf("Гипотеза подтверждена\n");
        }else{
            printf("X2 > XiTable\n");
```

```
            printf("Гипотеза опровергнута\n");
        }
    }else{
        printf("а не найдена в таблице или количество интервалов/точек меньше 4\n");
    }
}

int
main(void)
{
    struct dpoint mas[100];
    printf("0 - ввод из файла\n");
    printf("1 - ручной ввод\n");
    int tmp;
    int tmp1;
    int tmp2;
    int n = 0;
    scanf("%d",&tmp1);
    printf("0 - массив данных\n");
    printf("1 - интервальный ряд\n");
    scanf("%d",&tmp2);
    if(tmp1){
        if(tmp2){
            printf("Введите количество интервалов\n");
        }else{
            printf("Введите количество чисел\n");
        }
        scanf("%d",&n);
        n = HandleInput(mas,n,tmp2);
    }else{
        n = generateMas(mas,"data.txt",tmp2);
    }
    sort(mas,n);
    printf("N is %d\n",n);
    printf("Проверка гипотезы о виде распределения по критерию Пирсона\n");
    printf("0 - о нормальном распределении\n");
    printf("1 - о биномиальном распределении\n");
    scanf("%d",&tmp);
    printf("Введите a:\n");
    double a;
    scanf("%lf",&a);
    if (tmp){
        double cnt;
        printf("Введите N - число испытаний\n");
        scanf("%lf",&cnt);
        CheckBin(mas,n,a,cnt,tmp2);
    }else{
        CheckNorm(mas,n,a,tmp2);
    }
    printf("Введите целое число для завершения\n");
    scanf("%d",&n);
    // delData(tmp);
    return 0;
}
```