#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define row 181

#define col 217

void assign(float[],int);

double\*\*\*U;

main()

{

FILE \*fp1,\*fp2;

struct image

{ int intensity;

int cluster\_id;

}\*\*img;

float cent[10],temp[5],y[5],d,e,f,g,sum,max,num,den,diff,intra,min,inter,N,mul,VI,sum1,Vpc,Vpe,V;

int i,j,k=0,l,m,n,ctr=0,flag=0,pos;

fp1=fopen("1\_070.txt","r");

fp2=fopen("1\_070(FCM).pgm","w");

printf(" FCM \n");

printf("No. of cluster centres: ");

scanf("%d",&n);

printf("Enter the values of the cluster centres: ");

for(i=0;i<n;i++)

scanf("%f",&cent[i]);

//Memory Allocation

img = (struct image\*\*)calloc(row,sizeof(struct image\*));

for(i=0;i<row;i++)

img[i] = (struct image\*)calloc(col,sizeof(struct image));

U=(double\*\*\*)calloc(row,sizeof(double\*\*));

for(i=0;i<row;i++)

{

U[i]=(double\*\*)calloc(col,sizeof(double\*));

for(j=0;j<col;j++)

U[i][j]=(double\*)calloc(10,sizeof(double));

}

//Reading image

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

fscanf(fp1,"%d ",&img[i][j].intensity);

}

}

printf("\n------File read------\n");

while(1)

{

for(i=0;i<n;i++)

temp[i]=cent[i];

//Membership Matrix

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

for(k=0;k<n;k++)

{

sum=0;

d=fabs(img[i][j].intensity-cent[k]);

if(d==0)

U[i][j][k]=1;

else

{

for(m=0;m<n;m++)

{

e=fabs(img[i][j].intensity-cent[m]);

sum+=(d\*d)/(e\*e);

}

U[i][j][k]=1/sum;

}

}

}

}

//Allocation

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

max=0;

for(k=0;k<n;k++)

{

if(U[i][j][k]>max)

{

max=U[i][j][k];

pos=k;

}

}

img[i][j].cluster\_id=pos;

}

}

//Recalculation

for(k=0;k<n;k++)

{

num=0;

den=0;

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

if(img[i][j].cluster\_id==k)

{

num+=U[i][j][k]\*U[i][j][k]\*img[i][j].intensity;

den+=U[i][j][k]\*U[i][j][k];

}

}

}

cent[k]=num/den;

}

for(i=0;i<n;i++)

printf("%.2f ",cent[i]);

printf("\n");

ctr=0;

for(i=0;i<n;i++)

{

y[i]=fabs(temp[i]-cent[i]);

if(y[i]<=0.50)

ctr++;

}

if(ctr==n)

{

flag=1;

break;

}

}

//Validity Index

//sum=0;

sum1=0;

for(k=0;k<n;k++)

{

sum=0;

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

if(img[i][j].cluster\_id==k)

{

diff=fabs(img[i][j].intensity-cent[k]);

diff=diff\*diff;

sum=sum+diff;

}

}

}

sum1=sum1+sum;

}

intra=(sum1/(1.0\*row\*col));

min=99999;

for(i=0;i<n-1;i++)

{

for(j=i+1;j<n;j++)

{

diff=fabs(cent[i]-cent[j]);

printf("diff = %.2f\n",diff);

if((diff\*diff)<min)

min=diff\*diff;

}

//printf("min = %.2f\n",min);

}

inter=min;

d=pow((2\*3.14),0.5);

e=((n-2)\*(n-2))/2.0;

N=(1/(1.0\*d))\*exp(-e);

mul=(25\*N)+1;

VI=mul\*intra/inter;

printf("VI = %f\n",VI);

//assign(cent,n);

//Replacing the pixels with the centroid value

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

for(k=0;k<n;k++)

{

sum=0;

d=fabs(img[i][j].intensity-cent[k]);

if(d==0)

U[i][j][k]=1;

else

{

for(m=0;m<n;m++)

{

e=fabs(img[i][j].intensity-cent[m]);

sum+=(d\*d)/(e\*e);

}

U[i][j][k]=1/sum;

}

}

}

}

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

max=0;

for(k=0;k<n;k++)

{

if(U[i][j][k]>max)

{

max=U[i][j][k];

pos = k;

}

}

img[i][j].intensity = cent[pos];

}

}

//Writing into file

fprintf(fp2,"P2 %d %d 255 ",row,col);

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

fprintf(fp2,"%d ",img[i][j].intensity);

}

}

void assign(float cent[],int n)

{

int i,j;

double temp;

for(i=0;i<n;i++)

{

for(j=0;j<n-i-1;j++)

{

if(cent[j]>cent[j+1])

{

temp=cent[j];

cent[j]=cent[j+1];

cent[j+1]=temp;

}

}

}

for(i=0;i<n;i++)

{

cent[i]=(i/(n-1))\*255;

}

}