import java.io.\*;

import java.sql.\*;

public class ExtractionF

{

public static void main(String args[]) throws IOException,SQLException

{

BufferedWriter bw=new BufferedWriter(new FileWriter("region.txt"));

BufferedReader br=new BufferedReader(new FileReader("region1.txt"));

Connection con=null;

Statement stmt=null;

ResultSet rs=null;

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dwm33");

stmt=con.createStatement();

}

catch(Exception e)

{

System.out.println(e);

}

String x;

x=br.readLine();

while(x!=null) //Normal Extraction

{

System.out.println(x);

bw.write(x);

bw.newLine();

x=br.readLine();

}

//-----------------------------------------------------------------------------------

String s="select \* from Region;";

rs=stmt.executeQuery(s);

rs.next(); //MDB Extraction

String str="",y="";

for(int i=0;i<8;i++)

{

for(int j=1;j<6;j++)

{

y=rs.getString(j);

if(j<5)

str+=y+",";

else

str+=y+";";

}

bw.write(str);

bw.newLine();

str="";

rs.next();

}

con.close();

}

}

OUTPUT:

Region.txt:

1,1,himalayas,M,1;

2,2,Western Ghats,M,2;

3,3,Eastern Ghats,M,3;

4,4,Sunderban,W,4;

5,5,Thar Desert,D,5;

6,6,Ganges,P,6;

7,7,Gir,F,7;

8,8,Jim Corbett,F,8;

1,1,himalayas,M,1;

2,2,Western Ghats,M,2;

3,3,Eastern Ghats,M,3;

4,4,Sunderban,W,4;

5,5,Thar Desert,D,5;

6,6,Ganges,P,6;

7,7,Gir,F,7;

8,8,Jim Corbett,F,8;

import java.io.\*;

import java.sql.\*;

public class Transformations

{

public static void main(String args[]) throws IOException,SQLException

{

BufferedWriter bw=new BufferedWriter(new FileWriter("region.txt"));

BufferedReader br=new BufferedReader(new FileReader("region1.txt"));

BufferedReader br2=new BufferedReader(new FileReader("region1.txt"));

BufferedReader br1=new BufferedReader(new FileReader("plant.txt"));

BufferedReader br3=new BufferedReader(new FileReader("split1.txt"));

BufferedReader br4=new BufferedReader(new FileReader("missing.txt"));

BufferedReader br5=new BufferedReader(new FileReader("regiondedup1.txt"));

Connection con=null;

Statement stmt=null;

ResultSet rs=null;

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dwm33");

stmt=con.createStatement();

}

catch(Exception e)

{

System.out.println(e);

}

//-----------------------------------------------------------------------------------

String x="";

String s="select \* from Region;";

rs=stmt.executeQuery(s);

rs.next(); //MDB Extraction

String str="",y="";

for(int i=0;i<8;i++)

{

for(int j=1;j<6;j++)

{

y=rs.getString(j);

if(j<5)

str+=y+",";

else

str+=y+";";

}

bw.write(str);

bw.newLine();

str="";

rs.next();

}

//----------------------------------------------------------------------------------

x=br2.readLine();

String formatted="",str2="";

int count=0,index=0;

while(x!=null) //Formatted Extraction

{

count++;

str2=x.replaceAll("mountains","M");

str2=str2.replaceAll("desert","D");

str2=str2.replaceAll("wetlands","W");

str2=str2.replaceAll("plains","P");

str2=str2.replaceAll("forests","F");

index=str2.indexOf(',');

formatted+=count+""+str2.substring(index,str2.length());

System.out.println(formatted);

bw.write(formatted);

bw.newLine();

x=br2.readLine();

formatted="";

}

br2.close();

//-----------------------------------------------------------------------------------------------

y=br1.readLine();

int firstcoma=0,secondcoma=0,thirdcoma=0;

int avgpop=0;

String pop2="",pop7="",pop12="";

while(y!=null) //Average plant pop calculation

{

firstcoma=findcoma(y,1);

secondcoma=findcoma(y,2);

thirdcoma=findcoma(y,3);

pop2=y.substring(firstcoma+1,secondcoma);

pop7=y.substring(secondcoma+1,thirdcoma);

pop12=y.substring(thirdcoma+1,y.indexOf(';'));

avgpop=(Integer.parseInt(pop2)+Integer.parseInt(pop7)+Integer.parseInt(pop12))/3;

formatted=y.substring(0,y.indexOf(';'))+","+avgpop+";";

bw.write(formatted);

bw.newLine();

formatted="";

y=br1.readLine();

}

br1.close();

//---------------------------------------------------------------------------------------------------------

y=br3.readLine(); //Splitting

while(y!=null)

{

secondcoma=findcoma(y,2);

thirdcoma=findcoma(y,3);

String format=y.substring(0,secondcoma);

x=y.substring(secondcoma,thirdcoma);

System.out.println(x);

x=x.replace(" ",",");

System.out.println(x);

format+=x+y.substring(thirdcoma,y.length()-1);

System.out.println(format);

bw.write(format);

format="";

bw.newLine();

y=br3.readLine();

}

br3.close();

//--------------------------------------------------------------------------------------------------------

x=br4.readLine(); //Missing

String str3="",str4="";

int c2=0,c3=0,c4=0;

formatted="";

index=0;count=0; str2="";

while(x!=null)

{

index=x.indexOf('$');

if(index==-1)

{

bw.write(x);

formatted="";

bw.newLine();

x=br4.readLine();

}

else

{

formatted+=x.substring(0,index-1);

c2=findcoma(x,2);

c3=findcoma(x,3);

c4=findcoma(x,4);

str3=x.substring(c4,x.length());

str2=x.substring(c2+1,c3);

System.out.println(str2);

if(str2.equals("mammal"))

str4+=",deer";

if(str2.equals("bird"))

str4+=",fishes and frogs";

if(str2.equals("reptile"))

str4+=",insects and frogs";

if(str2.equals("fish"))

str4+=",small fishes";

if(str2.equals("amphibian"))

str4+=",fishes";

formatted+=str4+str3;

str4="";

str3="";

bw.write(formatted);

formatted="";

bw.newLine();

x=br4.readLine();

}

}

//--------------------------------------------------------------------------------------------------------

x=br5.readLine(); //Deduplication

while(x!=null)

{

int t=0;

int v=0;

s="select \* from Region;";

rs=stmt.executeQuery(s);

rs.next();

for(int i=0;i<8;i++)

{

str="";y="";

for(int j=2;j<6;j++)

{

y=rs.getString(j);

if(j<5)

str+=y+",";

else

str+=y+";";

}

rs.next();

char xch[] = x.toCharArray();

int k ;

char xch2[] =new char[xch.length];

for( k=0;k<xch.length;k++)

{

if(xch[k]==',')

{

k++;

int c=k;

for(t=0;t<((xch.length)-k);t++)

{

xch2[t]=xch[c];

c++;

}

break;

}

}

char xch3[] =new char[t];

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

xch3[k]=xch2[k];

String x2 = new String(xch3);

v=str.compareTo(x2);

if(v==0)

{

break;

}

}

if(v!=0)

{

bw.write(x);

bw.newLine();

System.out.println(x);

}

x=br5.readLine();

}

bw.close();

br5.close();

}

//---------------------------------------------------------------------------------------------------------

static int findcoma(String x,int num) //Finding index of comas

{

int count=0,i=0;

while(x!=null)

{

if(x.charAt(i)==',')

count++;

if(count==num)

return i;

i++;

}

return i;

}

}

Output:

1,1,himalayan region,mountains,1;

2,2,Western Ghats,mountains,2;

3,3,Eastern Ghats,mountains,3;

4,4,Sunderban delta,wetlands,4;

5,5,Thar Desert,desert,5;

6,6,Ganges plain,plains,6;

7,7,Gir forest,forests,7;

8,8,Jim Corbett,forests,8;

1,1,himalayas,M,1;

2,2,Western Ghats,M,2;

3,3,Eastern Ghats,M,3;

4,4,Sunderban,W,4;

5,5,Thar Desert,D,5;

6,6,Ganges,P,6;

7,7,Gir,F,7;

8,8,Jim Corbett,F,8;

1,1000,1200,1100,1100;

2,900,645,456,667;

3,800,758,696,751;

4,450,450,450,450;

5,600,650,670,640;

6,750,782,790,774;

7,1500,1300,1000,1266;

8,540,540,540,540;

1,1,himalayan,region,mountains,1

2,2,Western,Ghats,mountains,2

3,3,Eastern,Ghats,mountains,3

4,4,Sunderban,delta,wetlands,4

5,5,Thar,Desert,desert,5

6,6,Ganges,plain,plains,6

7,7,Gir,forest,forests,7

8,8,Jim,Corbett,forests,8

1,black bears,mammal,deer,1;

2,himalayan tahr,mammal,deer,2;

3,himalayan yak,mammal,herbs,3;

4,golden eagle,bird,fishes and frogs,4;

5,asiatic lion,mammal,deer,rabbits,5;

6,indian cobras,reptile,insects and frogs,6;

7,hyenas,bird,carrion,birds,reptiles,7;

8,jackals,mammal,rabbits,deer,rats,8;

1,1,himalayas,mountains,1;

4,4,Sunderban,wetlands,4;

6,6,Ganges,plains,6;

7,7,Gir,forests,7;

10,10,tt,forests,10;

import java.io.\*;

import java.sql.\*;

public class KmeanSpecies

{

public static void main(String args[]) throws IOException,SQLException

{

BufferedWriter bw=new BufferedWriter(new FileWriter("regiondedup.txt"));

BufferedReader br=new BufferedReader(new FileReader("regiondedup1.txt"));

BufferedWriter bw1=new BufferedWriter(new FileWriter("finaldedup.txt"));

Connection con=null;

Statement stmt=null;

ResultSet rs=null;

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:dwm33");

stmt=con.createStatement();

}

catch(Exception e)

{

System.out.println(e);

}

int arr[]=new int[20];

int k1[]=new int[20];

int k2[]=new int[20];

int j=0,k=0;

float m1,m2;

float tem1,tem2;

int n=0,i;

String s="select avgspop from Species\_pop;";

rs=stmt.executeQuery(s);

rs.next(); //MDB Extraction

String str="",y="";

for(i=0;i<8;i++)

{

y=rs.getString(1);

int w=Integer.parseInt(y);

arr[i]=w;

n++;

y="";

rs.next();

}

bw.close();

m1=arr[1];m2=arr[5];

do

{

j=0;

k=0;

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

{

if(Math.abs(m1-arr[i])< Math.abs(m2-arr[i]))

k1[j++]=arr[i];

else

k2[k++]=arr[i];

}

System.out.println("\n m1\t m2\t k1\t\t k2");

System.out.println("\n" +m1+"\t"+m2 +"\t");

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

System.out.print(k1[i]+" ");

System.out.println("\t");

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

System.out.print(k2[i]+" ");

tem1=m1;tem2=m2;

int s1=0,s2=0;

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

{

s1=s1+ k1[i];

m1=s1/j;

}

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

{

s2=s2+ k2[i];

m2=s2/k;

}

if((tem1==m1)&&(tem2==m2))

break;

}while(true);

}

}

import java.io.\*;

class knn

{

public static void main(String arg[])throws IOException

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int a[]={100,700,140,210,350,400,550,160,870,220};

int cls[]={0,1,0,0,1,1,1,0,1,0};

int diff[]=new int[10];

System.out.println("Enter the new value");

int p=Integer.parseInt(br.readLine());

for(int i=0;i<10;i++)

{

diff[i]=a[i]-p;

if(diff[i]<0)

diff[i]=diff[i]\*(-1);

}

for(int i=0;i<10;i++)

{

for(int j=0;j<9;j++)

{

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

{

int t=diff[j];

diff[j]=diff[j+1];

diff[j+1]=t;

t=cls[j];

cls[j]=cls[j+1];

cls[j+1]=t;

}

}

}

for(int i=0;i<10;i++)

{

System.out.println(a[i]);

}

for(int i=0;i<10;i++)

{

System.out.println(cls[i]);

}

int c0=0,c1=0;

for(int i=0;i<5;i++)

{

if(cls[i]==0)

c0++;

else

c1++;

}

if(c1>c0)

System.out.println("Densely Populated");

else

System.out.println("Sparsely Populated");

}

}