8.2

import java.util.\*;

public class First{

public static void main(String args[]){

System.out.println("Enter a 4-by-4 matrix row by now:");

double[][] array=new double[4][4];

double sum=sumDiagonal(array);

System.out.println("The sum of teh elements in the major diagonal is"+sum);

}

public static double sumDiagonal(double[][] array){

Scanner ss=new Scanner(System.in);

double sum=0;

for(int m=0;m<array.length;m++)

{

for(int i=0;i<array[0].length;i++)

{

array[m][i]=ss.nextDouble();

if(m==i)

{

sum+=array[m][i];

}

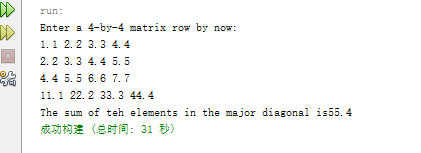
}

}

return sum;

}

}



8.3

源代码：

public class First{

public static void main(String args[]){

int[] rank1=new int[8];

int[] rank2=new int[8];

int[] human={0,0,0,0,0,0,0,0};

char[][] answers={

{'A','B','A','C','C','D','E','E','A','D'},

{'D','B','A','B','C','A','E','E','A','D'},

{'E','D','D','A','C','B','E','E','A','D'},

{'C','B','A','E','D','C','E','E','A','D'},

{'A','B','D','C','C','D','E','E','A','D'},

{'B','B','E','C','C','D','E','E','A','D'},

{'B','B','A','C','C','D','E','E','A','D'},

{'E','B','E','C','C','D','E','E','A','D'}};

char[] keys={'D','B','D','C','C','D','A','E','A','D'};

for(int i=0;i<answers.length;i++){

int correctCount=0;

for(int j=0;j<answers[i].length;j++){

if(answers[i][j]==keys[j])

correctCount++;

}

rank1[i]=correctCount;

rank2[i]=correctCount;

}

java.util.Arrays.sort(rank2);

for(int k=0;k<rank1.length;k++){

for(int m=0;m<rank2.length;m++){

if(rank1[m]==rank2[k]){

if(human[m]==0){

System.out.println("Student "+m+" 's correct count is "+rank2[k]);

human[m]++;

break;

}

else

continue;

}

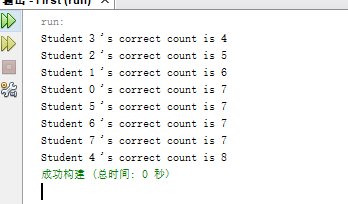
}

}

}

}

运行结果:



8.7

源代码：

public class First{

public static void main(String args[]){

System.out.println("Enter the number of point:");

double[][] points={{-1,0,3},{-1,-1,-1},{4,1,1},{2,0.5,9},{3.5,2,-1},{3,1.5,3},{-1.5,4,2},{5.5,4,-0.5}};

int p1=0,p2=1;

double shortestDistance=distance(points[p1][0],points[p1][1],points[p1][2],points[p2][0],points[p2][1],points[p2][2]);

for(int i=0;i<points.length;i++){

for(int j=i+1;j<points.length;j++){

double distance=distance(points[i][0],points[i][1],points[i][2],points[j][0],points[j][1],points[j][2]);

if(shortestDistance>distance){

p1=i;p2=j;

shortestDistance=distance;

}

}

}

System.out.println("The closest two points are "+"("+points[p1][0]+","+points[p1][1]+","+points[p1][2]+") and ("+points[p2][0]+","+points[p2][1]+","+points[p2][2]+")");

}

public static double distance(double x1,double y1,double z1,double x2,double y2,double z2){

return Math.sqrt(Math.pow(x2-x1,2)+Math.pow(y2-y1,2)+Math.pow(z2-z1,2));

}

}

运行结果：

