**算法训练 蜜蜂飞舞**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　“两只小蜜蜂呀，飞在花丛中呀……”  
  
　　话说这天天上飞舞着两只蜜蜂，它们在跳一种奇怪的舞蹈。用一个空间直角坐标系来描述这个世界，那么这两只蜜蜂初始坐标分别为(x1,y1,z1)，(x2,y2,z2)　　。在接下来它们将进行n次飞行，第i次飞行两只蜜蜂分别按照各自的速度向量飞行ti个单位时间。对于这一现象，玮玮已经观察了很久。他很想知道在蜜蜂飞舞结束时，两只蜜蜂的距离是多少。现在他就求教于你，请你写一个程序来帮他计算这个结果。

输入格式

　　第一行有且仅有一个整数n，表示两只蜜蜂将进行n次飞行。  
  
　　接下来有n行。  
  
　　第i行有7个用空格分隔开的整数ai,bi,ci,di,ei,fi,ti　　，表示第一只蜜蜂单位时间的速度向量为(ai,bi,ci) ，第二只蜜蜂单位时间的速度向量为(di,ei,fi) ，它们飞行的时间为ti 。  
  
　　最后一行有6个用空格分隔开的整数x1,y1,z1,x2,y2,z2，如题所示表示两只蜜蜂的初始坐标。

输出格式

　　输出仅包含一行，表示最后两只蜜蜂之间的距离。保留4位小数位。

样例输入

Sample 1

1

1 1 1 1 -1 1 2

3 0 1 2 0 0

Sample 2

3

1 1 1 1 -1 1 2

2 1 2 0 -1 -1 2

2 0 0 -1 1 1 3

3 0 1 2 0 0

样例输出

Sample 1

4.2426

Sample 2

15.3948

本题的C++参考代码如下：

#include<cstdio>

#include<cmath>

using namespace std;

int main(){

int x1=0,y1=0,z1=0,x2=0,y2=0,z2=0;

int a1,b1,c1,a2,b2,c2;

int t;

int n;

double s;

scanf("%d",&n);

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

scanf("%d%d%d%d%d%d%d",&a1,&b1,&c1,&a2,&b2,&c2,&t);

x1+=a1\*t;

x2+=a2\*t;

y1+=b1\*t;

y2+=b2\*t;

z1+=c1\*t;

z2+=c2\*t;

}

scanf("%d%d%d%d%d%d",&a1,&b1,&c1,&a2,&b2,&c2);

x1+=a1;

x2+=a2;

y1+=b1;

y2+=b2;

z1+=c1;

z2+=c2;

s=sqrt((double)(abs(x2-x1)\*abs(x2-x1)+abs(y2-y1)\*abs(y2-y1)+abs(z2-z1)\*abs(z2-z1)));

printf("%.4lf\n",s);

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#include <math.h>

int main()

{

int n;

double x1=0,y1=0,z1=0,x2=0,y2=0,z2=0;

int ai,bi,ci,di,ei,fi,ti;

int i;

scanf("%d",&n);

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

{

scanf("%d%d%d%d%d%d%d",&ai,&bi,&ci,&di,&ei,&fi,&ti);

x1+=ai\*ti;

y1+=bi\*ti;

z1+=ci\*ti;

x2+=di\*ti;

y2+=ei\*ti;

z2+=fi\*ti;

}

scanf("%d%d%d%d%d%d",&ai,&bi,&ci,&di,&ei,&fi);

x1+=ai;

y1+=bi;

z1+=ci;

x2+=di;

y2+=ei;

z2+=fi;

printf("%.4lf",sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2)+(z1-z2)\*(z1-z2)));

return 0;

}

本题的Java参考代码如下：

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int x1 = 0, y1 = 0, z1 = 0;

int x2 = 0, y2 = 0, z2 = 0;

for (; n > 0; --n) {

int a1 = sc.nextInt(), b1 = sc.nextInt(), c1 = sc.nextInt();

int a2 = sc.nextInt(), b2 = sc.nextInt(), c2 = sc.nextInt();

int t = sc.nextInt();

x1 += a1 \* t;

y1 += b1 \* t;

z1 += c1 \* t;

x2 += a2 \* t;

y2 += b2 \* t;

z2 += c2 \* t;

}

x1 += sc.nextInt();

y1 += sc.nextInt();

z1 += sc.nextInt();

x2 += sc.nextInt();

y2 += sc.nextInt();

z2 += sc.nextInt();

x1 -= x2;

y1 -= y2;

z1 -= z2;

System.out.printf("%.4f", Math.sqrt(x1 \* x1 + y1 \* y1 + z1 \* z1));

}

}