**A卷**

**第一题 基本运算**

**题目描述：**

输入10个整数，求出这些整数的最大值、最小值，以及它们的和。

输入一个由字母和数字组成的字符串（长度<=20），将其中的大写字母转换成小写字母后输出该字符串。

**输入：**

第一行10个整数，用空格分隔。第二行一个由字母和数字组成的字符串，长度不超过20个字符。

**输出：**

两行，第一行最大值、最小值、和，用空格分隔，行尾无空格；第二行大写字母转成小写字母后的字符串。

**样例输入：**

1 2 3 4 5 6 7 8 9 0

abc123BCD

**样例输出：**

9 0 45

abc123bcd

**参考答案：**

## ****第二题 矩阵运算****

### ****题目描述：****

给定两个m阶矩阵（m<=10）A和B，按照如下要求对矩阵进行处理后得出运算结果。

1、 对矩阵A和B中的素数元素加1，再求矩阵A和B的和C1，即C1=A+B。

2、 对矩阵A和B中的奇数元素减1，再求矩阵A和B的差C2，即C2=A-B。

3、 对矩阵A和B中的偶数元素乘2，再求矩阵A和B的积C3，即C3=A\*B。

### ****输入：****

第一行一个整数m（m<=10），代表矩阵A和B的阶数。接下来有2\*m行，每行有m个整数，整数之间用空格分隔，分别代表矩阵A和B。

### ****输出：****

按照题目要求输出运算后的矩阵C1、C2、C3，共3\*m行，每行m个元素，元素之间用空格分隔，行尾无空格。

### ****样例输入：****

**样例输入：**

3

1 2 3

4 5 6

7 8 9

2 3 4

5 6 7

8 9 10

**样例输出：**

4 7 8

10 12 14

16 17 19

-2 0 -2

0 -2 0

-2 0 -2

72 78 96

249 192 339

252 294 348

**参考答案**

**第三题 函数**

**题目描述**

勇士林克在哈特诺村服饰染色点店老板瑟吉处得知，付款20卢比的同时还得寻找各种颜色的材料才能进行染色，林克决定自行研究新的佐纳乌科技进行服饰染色。

现已知佐纳乌装置需要一个十六进制的颜色代码进行染色，如”#00ffff”表示青色。但不同场景中表示颜色方式有很多种，如RGB色彩模式、CMYK印刷四分色模式。请帮助林克写一个重载函数Color2Hex()来完成不同形式颜色数据向十六进制颜色代码的转换。

RGB色彩模式：常用的颜色模式，有三个整数类型颜色分量R（Red红色）G（Green绿色）B（Blue蓝色），每个分量为0到255的整数，如（0,255,255）就表示青色；每个分量转换成十六进制数拼接起来，（0,255,255）对应的十六进制颜色代码就为”#00ffff”。

CMYK四分色模式：三原混色加上黑色的印刷套色模式，有四个分量C（Cyan青色）M（Magenta品红色）Y（Yellow黄色）K（blacK表示灰度），每个分量为0到1之前的浮点数，如（1.0000, 0.0000, 0.0000, 0.0000）表示青色。CMYK转换RGB的计算公式如下：

R = (int) 255.0\*(1-C)\*(1-K)；

G = (int) 255.0\*(1-M)\*(1-K)；

B = (int) 255.0\*(1-Y)\*(1-K)；

请完成下面的函数功能：

Void Color2Hex(int r, int g, int b);// 实现RGB数据向十六进制颜色代码转换，并输出颜色代码

Void Color2Hex(float c, float m, float y, float k); // 实现CMYK数据向十六进制颜色代码转换，并输出颜色代码

**输入:**

程序接受两行输入：第一行为一个字符，r（或R）表示第二行为RGB数据， c（或C）表示第二行为CMYK数据；第二行为与第一行对应的输入数据。

**输出:**

针对输入数据进行转换后的十六进制颜色代码，格式为”#xxxxxx”，x为对应的十六进制符号(0~f)。如果输入的字符错误或者输入的数据不规范，输出“Wrong Input!”。

**样例输入**

C

1.0 0.0 0.0 0.0

**样例输出**

#00ffff

**参考答案**

**第四题 结构体**

**题目描述**

请实现一个基础类型坐标点结构体point，结构体成员包含x轴、y轴坐标x、y。

1. 实现一个直线结构体line，结构体成员包含直线上两个点的坐标 point a、point b 。现要求依次输入两个直线的四个点坐标，请判断两个直线是否平行，若平行输出“True”，否则输出“False”。需要明确的是，不考虑直线重叠的情况。
2. 实现一个矩形结构体rect，结构体成员包含左下角和右上角的两个点坐标 point a、point b 。现要求依次输入两个矩形的四个点坐标，请判断两个矩形是否重叠，若重叠输出两个矩形重叠部分形成矩形的面积，否则输出“False”。需要明确的是，矩形的边与坐标轴垂直/平行，只在角或边接触的两个矩形不构成重叠。

注意，本题必须使用结构体实现，否则记零分。

**输入**

x1 y1 x2 y2

x3 y3 x4 y4

x5 y5 x6 y6

x7 y7 x8 y8

其中x1 y1 x2 y2为第一个直线的点坐标，x3 y3 x4 y4为第二个直线的点坐标，均为整数。

其中x5 y5 x6 y6为第一个矩形的点坐标，x7 y7 x8 y8为第二个矩形的点坐标，均为整数。

**输出**

True / False

area / False

**样例输入**

0 0 0 1

1 0 1 1

0 0 2 2

1 1 3 3

**样例输出**

True

1

**参考答案**

**B卷**

**第一题 基本运算**

**题目描述：**

输入10个整数，求出这些整数的最小值、最大值，以及它们的和。

输入一个由字母和数字组成的字符串（长度<=20），将其中的小写字母转换成大写字母后输出该字符串。

**输入：**

第一行10个整数，用空格分隔。第二行一个由字母和数字组成的字符串，长度不超过20个字符。

**输出：**

两行，第一行最小值、最大值、和，用空格分隔，行尾无空格；第二行小写字母转成大写字母后的字符串。

**样例输入：**

1 2 3 4 5 6 7 8 9 0

abc123BCD

**样例输出：**

0 9 45

ABC123BCD

**参考答案**

#include<iostream>

#include<cstring>

#include<iomanip>

#include<cmath>

#include<algorithm>

using namespace std;

int main(){

int arr[15]={0};

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

cin>>arr[i];

int maxx=arr[0],minn=arr[0],sum=0;

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

{

if(maxx<arr[i])

maxx=arr[i];

if(minn>arr[i])

minn=arr[i];

sum+=arr[i];

}

char a[25];

cin>>a;

for(size\_t j=0;j<strlen(a);j++)

{

if(a[j]>='a'&&a[j]<='z')

a[j]=(char)(a[j]+'A'-'a');

}

cout<<minn<<" "<<maxx<<" "<<sum<<endl;

for(size\_t j=0;j<strlen(a);j++)

cout<<a[j];

return 0;

}

#include<iostream>

using namespace std;

#include<string>

int main()

{

int suma=0;

int arry[10];

string a;

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

{

cin>>arry[i];

suma+=arry[i];

}

cin>>a;

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

{

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

{

if(arry[i]>arry[j])

{

int t=arry[i];

arry[i]=arry[j];

arry[j]=t;

}

}

}

cout<<arry[0]<<" "<<arry[9]<<" "<<suma<<endl;

for(int i=0;i<a.size();i++)

{

if(a[i]>='a'&&a[i]<='z')

{

cout<<char(a[i]-32);

}

else

{

cout<<a[i];

}

}

return 0;

}

**第二题 矩阵运算**

**题目描述：**

输入10个整数，求出这些整数的最小值、最大值，以及它们的和。

输入一个由字母和数字组成的字符串（长度<=20），将其中的小写字母转换成大写字母后输出该字符串。

**输入：**

第一行10个整数，用空格分隔。第二行一个由字母和数字组成的字符串，长度不超过20个字符。

**输出：**

两行，第一行最小值、最大值、和，用空格分隔，行尾无空格；第二行小写字母转成大写字母后的字符串。

**样例输入：**

1 2 3 4 5 6 7 8 9 0

abc123BCD

**样例输出：**

0 9 45

**参考答案**

#include<iostream>

#include<cstring>

#include<iomanip>

#include<cmath>

#include<algorithm>

using namespace std;

long long n,ss;

bool judge(long long a)

{

if(a<=1)

return false;

if(a==2)return true;

else

for(int i=2;i<a/2+1;i++)

{

if(a%i==0)

return false;

}

return true;

}

void one(long long a[12][12],long long b[12][12])

{

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

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

{

if(!judge(a[i][j]))

{

//cout<<a[i][j]<<endl;

a[i][j]+=1;

//cout<<a[i][j]<<" ";

}

if(!judge(b[i][j]))

{

b[i][j]+=1;

}

}

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

{

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

cout<<a[i][j]+b[i][j]<<" ";

}

cout<<endl;

}

}

void two(long long a[12][12],long long b[12][12])

{

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

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

{

if(a[i][j]%2==0)

{a[i][j]-=1;}

if(b[i][j]%2==0)

{b[i][j]-=1;}

}

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

{

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

cout<<a[i][j]-b[i][j]<<" ";

}

cout<<endl;

}

}

void three(long long a[12][12],long long b[12][12])

{

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

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

{

if(a[i][j]%2!=0)

{

a[i][j]\*=2;

}

if(b[i][j]%2!=0)

{

b[i][j]\*=2;

}

}

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

{

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

{

for(int s1=0;s1<n;s1++)

ss+=a[i][s1]\*b[s1][j];

cout<<ss<<" ";

ss=0;

}

cout<<endl;

}

}

int main(){

cin>>n;

long long A[12][12]={0};

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

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

cin>>A[i][j];

long long B[12][12]={0};

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

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

cin>>B[i][j];

long long C[12][12],D[12][12],E[12][12],F[12][12];

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

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

{C[i][j]=A[i][j];E[i][j]=A[i][j];}

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

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

{D[i][j]=B[i][j];F[i][j]=B[i][j];}

one(A,B);

two(C,D);

three(E,F);

return 0;

}

#include<iostream>

using namespace std;

int A[10][10];

int B[10][10];

int c1[10][10];

int c2[10][10];

int c3[10][10];

int su(int n)

{

if(n==1||n<=0)

{

return 0;

}

else if(n==2)

{

return 1;

}

else

{

for(int i =2;i<n;i++)

{

if(n%i==0)

{

return 0;

}

}

return 1;

}

}

int main()

{

int m;

cin>>m;

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

{

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

{

cin>>A[i][j];

}

}

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

{

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

{

cin>>B[i][j];

}

}

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

{

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

{

if(su(A[i][j]))

{

c1[i][j]+=A[i][j];

}

else

{

c1[i][j]+=(A[i][j]+1);

}

if(su(B[i][j]))

{

c1[i][j]+=B[i][j];

}

else

{

c1[i][j]+=(B[i][j]+1);

}

}

}

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

{

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

{

if(A[i][j]%2!=0)

{

c2[i][j]+=A[i][j];

}

else

{

c2[i][j]+=(A[i][j]-1);

}

if(B[i][j]%2!=0)

{

c2[i][j]-=B[i][j];

}

else

{

c2[i][j]-=(B[i][j]-1);

}

}

}

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

{

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

{

if(A[i][j]%2!=0)

{

A[i][j]\*=2;

}

if(B[i][j]%2!=0)

{

B[i][j]\*=2;

}

}

}

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

{

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

{

for(int k=0;k<m;k++)

{

c3[i][j]+=A[i][k]\*B[k][j];

}

}

}

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

{

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

{

cout<<c1[i][j]<<" ";

}

cout<<endl;

}

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

{

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

{

cout<<c2[i][j]<<" ";

}

cout<<endl;

}

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

{

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

{

cout<<c3[i][j]<<" ";

}

cout<<endl;

}

return 0;

}

**第三题 函数**

**题目描述**

勇士林克在哈特诺村服饰染色点店老板瑟吉处得知，付款20卢比的同时还得寻找各种颜色的材料才能进行染色，林克决定自行研究新的佐纳乌科技进行服饰染色。

现已知佐纳乌装置需要一个十六进制的颜色代码进行染色，如”#00ffff”表示青色。但不同场景中表示颜色方式有很多种，如RGB色彩模式、CMYK印刷四分色模式和为了便于计算的十进制数形式。请帮助林克写一个重载函数Color2Hex()来完成不同形式颜色数据向十六进制颜色代码的转换。

RGB色彩模式：常用的颜色模式，有三个整数类型颜色分量R（Red红色）G（Green绿色）B（Blue蓝色），每个分量为0到255的整数，如（0,255,255）就表示青色；每个分量转换成十六进制数拼接起来，（0,255,255）对应的十六进制颜色代码就为”#00ffff”。

十进制数形式：在计算时偶尔使用的格式，就是把16进制数代码对应转成10进制数的结果，如65535就表示青色，对应的十六进制颜色代码就为”#00ffff”。

请完成下面的函数功能：

Void Color2Hex(int r, int g, int b);// 实现RGB数据向十六进制颜色代码转换，并输出颜色代码

Void Color2Hex(long num); //实现十进制数形式向十六进制颜色代码转换，并输出颜色代码

**输入:**

程序接受两行输入：第一行为一个字符，r（或R）表示第二行为RGB数据，d(或D)表示第二行为十进制数据；第二行为与第一行对应的输入数据。

**输出:**

针对输入数据进行转换后的十六进制颜色代码，格式为”#xxxxxx”，x为对应的十六进制符号(0~f)。如果输入的字符错误或者输入的数据不规范，输出“Wrong Input!”。

**样例输入**

d

65535

**样例输出**

#00ffff

**参考答案**

#include<iostream>

#include<cmath>

using namespace std;

bool isprime(int n)

{

if(n<=1)return false;

for(int i=2;i<n;i++)

{

if(n%i==0)return false;

}

return true;

}

bool func(int n)

{

if (abs(n)%2==0) return true;

return false;

}

int main()

{

int m;

cin>>m;

int A[11][11];

int B[11][11];

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

{

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

{

cin>>A[i][j];

}

}

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

{

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

{

cin>>B[i][j];

}

}

int A1[11][11],B1[11][11];

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

{

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

{

A1[i][j]=A[i][j];

B1[i][j]=B[i][j];

if(!isprime(A1[i][j]))A1[i][j]=A1[i][j]+1;

if(!isprime(B1[i][j]))B1[i][j]=B1[i][j]+1;

cout<<A1[i][j]+B1[i][j];

if(j!=m-1)cout<<" ";

}

cout<<endl;

}

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

{

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

{

A1[i][j]=A[i][j];

B1[i][j]=B[i][j];

if(func(A1[i][j]))A1[i][j]=A1[i][j]-1;

if(func(B1[i][j]))B1[i][j]=B1[i][j]-1;

cout<<A1[i][j]-B1[i][j];

if(j!=m-1)cout<<" ";

}

cout<<endl;

}

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

{

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

{

A1[i][j]=A[i][j];

B1[i][j]=B[i][j];

if(!func(A1[i][j]))A1[i][j]\*=2;

if(!func(B1[i][j]))B1[i][j]\*=2;

}

}

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

{

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

{

int result=0;

for(int k=0;k<m;k++)

{

result+=A1[i][k]\*B1[k][j];

}

cout<<result;

if(j!=m-1)cout<<" ";

}

cout<<endl;

}

//system("pause");

return 0;

}

#include<iostream>

using namespace std;

void Color2Hex(int r, int g, int b)

{

if(r<0||r>255||g<0||g>255||b<0||b>255)

{

cout<<"Wrong Input!";

return;

}

else

{

int k[7];

k[2]=r%16;

k[1]=r/16%16;

k[4]=g%16;

k[3]=g/16%16;

k[6]=b%16;

k[5]=b/16%16;

cout<<"#";

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

{

if(k[i]>9)

{

cout<<char(k[i]+87);

}

else

{

cout<<k[i];

}

}

return;

}

}

void Color2Hex(long num)

{

int k[7];

long long maxn=15;

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

{

maxn\*=16;

maxn+=15;

}

if(num>maxn||num<0)

{

cout<<"Wrong Input!";

return;

}

else

{

for(int i=6;i>0;i--)

{

k[i]=num%16;

num/=16;

}

cout<<"#";

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

{

if(k[i]>9)

{

cout<<char(k[i]+87);

}

else

{

cout<<k[i];

}

}

return;

}

}

int main()

{

char x;

cin>>x;

if(x=='r'||x=='R')

{

int r,g,b;

cin>>r>>g>>b;

Color2Hex(r,g,b);

}

else if(x=='d'||x=='D')

{

long num;

cin>>num;

Color2Hex(num);

}

else

{

cout<<"Wrong Input!";

}

return 0;

}

**第四题 结构体**

**题目描述**

请实现一个基础类型坐标点结构体point，结构体成员包含x轴、y轴坐标x、y。

1. 实现一个直线结构体line，结构体成员包含直线上两个点的坐标 point a、point b 。现要求依次输入两个直线的四个点坐标，请判断两个直线是否垂直，若垂直输出“True”，否则输出“False”。需要明确的是，不考虑直线重叠的情况。
2. 实现一个矩形结构体rect，结构体成员包含左下角和右上角的两个点坐标 point a、point b 。现要求依次输入两个矩形的四个点坐标，请判断两个矩形是否重叠，若重叠输出两个矩形重叠部分形成矩形的周长，否则输出“False”。需要明确的是，矩形的边与坐标轴垂直/平行，只在角或边接触的两个矩形不构成重叠。

注意，本题必须使用结构体实现，否则记零分。

**输入**

x1 y1 x2 y2

x3 y3 x4 y4

x5 y5 x6 y6

x7 y7 x8 y8

其中x1 y1 x2 y2为第一个直线的点坐标，x3 y3 x4 y4为第二个直线的点坐标，均为整数。

其中x5 y5 x6 y6为第一个矩形的点坐标，x7 y7 x8 y8为第二个矩形的点坐标，均为整数。

**输出**

True / False

perimeter / False

**样例输入**

0 0 0 1

0 1 1 1

0 0 2 2

1 1 3 3

**样例输出**

True

4

**参考答案**

#include<iostream>

#include<cstring>

#include<iomanip>

#include<cmath>

#include<algorithm>

using namespace std;

struct line

{

int x1;

int y1;

int x2;

int y2;

};

struct rect

{

int x1;

int y1;

int x2;

int y2;

};

int main() {

line a, b;

cin >> a.x1 >> a.y1 >> a.x2 >> a.y2;

cin >> b.x1 >> b.y1 >> b.x2 >> b.y2;

if ((a.x1 == a.x2 && b.y1 == b.y2) || (a.y1 == a.y2 && b.x1 == b.x2))

cout << "True" << endl;

else if ((a.y2 - a.y1) \* (b.y2 - b.y1) / (a.x2 - a.x1) / (b.x2 - b.x1) == -1)

cout << "True" << endl;

else

cout << "False" << endl;

rect c, d;

cin >> c.x1 >> c.y1 >> c.x2 >> c.y2;

cin >> d.x1 >> d.y1 >> d.x2 >> d.y2;

if (c.x1 == d.x1 && c.x2 == d.x2 && c.y1 == d.y1 && c.y2 == d.y2)

{

int ans = 0;

ans += 2 \* (max(c.y1, c.y2) - min(c.y1, c.y2)) + 2 \* (max(c.x1, c.x2) - min(c.x1, c.x2));

cout << ans;

}

else if (((d.x1 > min(c.x1, c.x2) && d.x1 < max(c.x1, c.x2)) && (d.y1 > min(c.y1, c.y2) && d.y1 < max(c.y1, c.y2))))

{

int ans = 0;

ans += 2 \* (max(c.x1, c.x2) - d.x1) + 2 \* (max(c.y1, c.y2) - d.y1);

cout << ans;

}

else if (((d.x2 > min(c.x1, c.x2) && d.x2 < max(c.x1, c.x2)) && (d.y2 > min(c.y1, c.y2) && d.y2 < max(c.y1, c.y2))))

{

int ans = 0;

ans += 2 \* (d.x2 - min(c.x1, c.x2)) + 2 \* (d.y2 - min(c.y1, c.y2));

cout << ans;

}

else if (((d.x2 > min(c.x1, c.x2) && d.x2 < max(c.x1, c.x2)) && (d.y1 > min(c.y1, c.y2) && d.y1 < max(c.y1, c.y2))))

{

int ans = 0;

ans += 2 \* (d.x2 - min(c.x2, c.x1) + 2 \* (max(c.y1, c.y2) - d.y1));

cout << ans;

}

else if (((d.x1 > min(c.x1, c.x2) && d.x1 < max(c.x1, c.x2)) && (d.y2 > min(c.y1, c.y2) && d.y2 < max(c.y1, c.y2))))

{

int ans = 0;

ans += 2 \* (max(c.x1, c.x2) - d.x1) + 2 \* (d.y2 - min(c.y1, c.y2));

cout << ans;

}

else

cout << "False";

return 0;

}

#include<iostream>

using namespace std;

struct point

{

double x, y;

};

struct line

{

point a;

point b;

};

struct rect

{

point a;

point b;

};

int main()

{

line l1;

line l2;

rect r1;

rect r2;

cin>>l1.a.x>>l1.a.y>>l1.b.x>>l1.b.y;

cin>>l2.a.x>>l2.a.y>>l2.b.x>>l2.b.y;

cin>>r1.a.x>>r1.a.y>>r1.b.x>>r1.b.y;

cin>>r2.a.x>>r2.a.y>>r2.b.x>>r2.b.y;

if(l1.a.x==l1.b.x&&l2.a.y==l2.b.y||l1.a.y==l1.b.y&&l2.a.x==l2.b.x)

{

cout<<"True"<<endl;

}

else

{

if(l1.a.x==l1.b.x||l2.a.x==l2.b.x)

{

cout<<"False"<<endl;

}

else

{

if((l1.a.y-l1.b.y)\*(l2.a.y-l2.b.y)/(l1.a.x-l1.b.x)/(l2.a.x-l2.b.x)==-1)

{

cout<<"True"<<endl;

}

else

{

cout<<"False"<<endl;

}

}

}

int sa1=r1.a.x+r1.a.y;

int sb1=r1.b.x+r1.b.y;

int sa2=r2.a.x+r2.a.y;

int sb2=r2.b.x+r2.b.y;

int s1max=(sa1>sb1?sa1:sb1);

int s1min=(sa1<sb1?sa1:sb1);

int s2max=(sa2>sb2?sa2:sb2);

int s2min=(sa2<sb2?sa2:sb2);

if(r1.b.x!=r2.a.x&&r1.b.y!=r2.a.y&&r2.b.x!=r1.a.x&&r2.b.y!=r1.a.y)

{

if(s1max>s2min&&s1min<s2min&&s1max<s2max)

{

cout<<2\*(s1max-s2min);

}

else if(s2max>s1min&&s2min<s1min&&s2max<s1max)

{

cout<<2\*(s2max-s1min);

}

else if(s1min<=s2min&&s1max>=s2max)

{

cout<<2\*(s2max-s2min);

}

else if(s2min<=s1min&&s2max>=s1max)

{

cout<<2\*(s1max-s1min);

}

else

{

cout<<"False";

}

}

else

{

cout<<"False";

}

return 0;

}

//矩阵重叠这个想法有问题，但是可以过80%的数据

#include<iostream>

using namespace std;

#include<cmath>

struct point

{

double x, y;

};

struct line

{

point a;

point b;

};

struct rect

{

point a;

point b;

};

int main()

{

line l1;

line l2;

rect r1;

rect r2;

cin>>l1.a.x>>l1.a.y>>l1.b.x>>l1.b.y;

cin>>l2.a.x>>l2.a.y>>l2.b.x>>l2.b.y;

cin>>r1.a.x>>r1.a.y>>r1.b.x>>r1.b.y;

cin>>r2.a.x>>r2.a.y>>r2.b.x>>r2.b.y;

if(l1.a.x==l1.b.x&&l2.a.y==l2.b.y||l1.a.y==l1.b.y&&l2.a.x==l2.b.x)

{

cout<<"True"<<endl;

}

else

{

if(l1.a.x==l1.b.x||l2.a.x==l2.b.x)

{

cout<<"False"<<endl;

}

else

{

if((l1.a.y-l1.b.y)\*(l2.a.y-l2.b.y)/(l1.a.x-l1.b.x)/(l2.a.x-l2.b.x)==-1)

{

cout<<"True"<<endl;

}

else

{

cout<<"False"<<endl;

}

}

}

int sa1=r1.a.x+r1.a.y;

int sb1=r1.b.x+r1.b.y;

int sa2=r2.a.x+r2.a.y;

int sb2=r2.b.x+r2.b.y;

if(r2.a.x>=r1.a.x&&r2.a.y>=r1.a.y&&r2.b.x<=r1.b.x&&r2.b.y<=r1.b.y)

{

cout<<2\*(sb2-sa2);

}

else if(r2.a.x<=r1.a.x&&r2.a.y<=r1.a.y&&r2.b.x>=r1.b.x&&r2.b.y>=r1.b.y)

{

cout<<2\*(sb1-sa1);

}

else if(r2.a.x>=r1.a.x&&r2.a.y>=r1.a.y&&r2.a.x<r1.b.x&&r2.a.y<r1.b.y)

{

int x1=min(r1.b.x,r2.b.x);

int y1=min(r1.b.y,r2.b.y);

cout<<2\*((x1-r2.a.x)+(y1-r2.a.y));

}

else if(r2.b.x<=r1.b.x&&r2.b.y<=r1.b.y&&r2.b.x>r1.a.x&&r2.b.y>r1.a.y)

{

int x1=max(r1.a.x,r2.a.x);

int y1=max(r1.a.y,r2.a.y);

cout<<2\*((r2.b.x-x1)+(r2.b.y-y1));

}

else

{

cout<<"False";

}

return 0;

}

//这是第二种思路，一些地方需要完善，可以过80%数据

**附加题 喜欢！雪！真实的魔法！**

*蓦然雪化,且静候,飞雪再临之时。*

**题目描述**

天津迎来了久违的大雪！在《White Album》的旋律中，Siannodel和鱼灰开始了校园内一条街道的扫雪工作。

整条街道长度为n，Siannodel和鱼灰各自有一个起始位置和恒定的速度，记为p1、v1和p2、v2.

他们可以在这段路上以任何路线进行扫雪工作，而只要其中一个人经过街道的某一位置，就可以扫清该位置上的积雪，当然，两个人可以先后经过同一位置，但也不会有额外的收获。

现在，请你为他们规划一个最佳的扫雪方案，使得扫清整条街道所有积雪的时间最短。

**输入**

本题有多组样例。

第一行输入一个实数t(0<t≤10000)，代表样例组数。

接下来的t行，每行输入五个数字：n(0<n≤10000)、p1(0≤p1≤n)、v1(0.001≤v1≤1000)、p2(0≤p2≤n、v2(0.001≤v2≤1000)。

**输出**

对于每个样例输出一行，包含一个数字，代表扫清积雪的最短时间。

若你的答案的绝对误差或相对误差不超过10−5，则被认为正确，因此请保留较多的小数位数。

**样例输入**

2

10000.0 1.0 0.001 9999.0 0.001

4306.063 4079.874 0.607 1033.423 0.847

**样例输出**

5001000.0000000000

3827.8370013755