#include <stdio.h>

int matrix[3][4]={0};

//将矩阵存入matrix

void get()

{

for(int i=0;i<3;++i)

for(int j=0;j<4;++j)

scanf("%d",&matrix[i][j]);

printf("\n");

}

//5.23

typedef struct Triple{

int j; //列

int data; //数据值

}Triple;

typedef struct row{

int i; //行

int index; //该行第一个非零元素在Triple的位置

}row;

void q23()

{

get();

//储存矩阵

int length=0;

Triple t[100];

row r[5];

for(int i=0;i<3;++i)

{

bool b=true; //判断是否为第一个元素

r[i].i=i;

for(int j=0;j<4;++j)

if(matrix[i][j]!=0)

{

++length;

t[length].j=j;

t[length].data=matrix[i][j];

if(b)

{

r[i].index=length;

b=false;

}

}

if(b) r[i].index=0; //第i行的值全为0

}

r[4].i=4;

r[4].index=length+1;

//查询元素

int x,y;

while(scanf("%d%d",&x,&y)!=EOF)

{

--x; --y;

int mint=r[x].index;

int maxt=r[x+1].index-1;

if(mint==0 || y<t[mint].j || y>t[maxt].j) //mint为0即为该行全为0的情况

{

printf("0\n"); continue;

}

bool b1=true;

for(int k=mint;k<=maxt;++k)

if(t[k].j==y)

{

printf("%d\n",t[k].data);

b1=false;

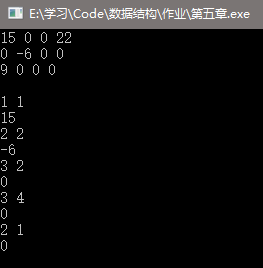
break;

}

if(b1) printf("0\n");

}

}



//5.25

int setV\_B(int\* V,int\* B) //设置V和B

{

get();

int count=0;

for(int i=0;i<3;++i)

for(int j=0;j<4;++j)

if(matrix[i][j]!=0)

{

V[count++]=matrix[i][j];

B[i\*4+j]=1;

}

return count;

}

void q25()

{

int V1[20],V2[20];

int B1[3][4]={0};

int B2[3][4]={0};

int count1=setV\_B(V1,\*B1);

int count2=setV\_B(V2,\*B2);

//矩阵相加

int C[3][4]={0};

int n1=0,n2=0;

for(int i=0;i<3;++i)

for(int j=0;j<4;++j)

{

int a=0,b=0;

if(B1[i][j]==1)

a=V1[n1++];

if(B2[i][j]==1)

b=V2[n2++];

C[i][j]=a+b;

}

//矩阵输出

for(int i=0;i<3;++i)

{

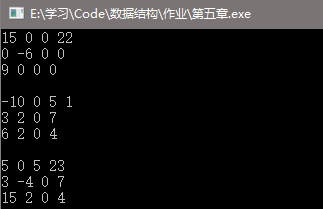
for(int j=0;j<4;++j)

printf("%d ",C[i][j]);

printf("\n");

}

}



int main()

{

// q23();

// q25();

return 0;

}