/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DO NOT EDIT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define true 1

#define false 0

typedef struct Points{

 float x;

 float y;

} Point;

typedef struct Rectangle{

 Point p1;

 Point p2;

 Point p3;

 Point p4;

} Rect;

void printPoint(Point p){

 printf("Point:: (%f, %f) \n", p.x, p.y);

}

void printRect(Rect r){

 printPoint(r.p1);

 printPoint(r.p2);

 printPoint(r.p3);

 printPoint(r.p4);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EDIT HERE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char collision(Rect r1, Rect r2){

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DO NOT EDIT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char test1(){

 Point p11; p11.x = 0; p11.y = 0;

 Point p12; p12.x = 1; p12.y = 0;

 Point p13; p13.x = 1; p13.y = 1;

 Point p14; p14.x = 0; p14.y = 1;

 Point p21; p21.x = 0.5; p21.y = 0.5;

 Point p22; p22.x = 3; p22.y = 0.5;

 Point p23; p23.x = 3; p23.y = 1.5;

 Point p24; p24.x = 0.5; p24.y = 1.5;

 Rect r1; r1.p1 = p11; r1.p2 = p12; r1.p3 = p13; r1.p4 = p14;

 Rect r2; r2.p1 = p21; r2.p2 = p22; r2.p3 = p23; r2.p4 = p24;

 return (collision(r1, r2) == false);

}

void main()

{

 if (test1())

 printf("Case 1 tested successfully");

 else

 printf("Case 1 test failed");

}