//------------------------------------------------------------------------------

/// \file intersection\_C\_File.c

//------------------------------------------------------------------------------

//

///

/// \brief To find whether 2 rectangles are intersecting

///

/// \n <b> Owner: Naveen Bolisetty

///

//------------------------------------------------------------------------------

// Externals

// none

// Header files

#include<stdio.h>

void intersection(int,int,int,int,int \*,int \*) ;

//------------------------------------------------------------------------------

/// \Parameters

//------------------------------------------------------------------------------

//

///  lenA\_start -- Starting point(X Axis- starting), of first rectangle

//   lenA\_end  --  Ending point(X Axis- starting), of first rectangle

//   lenB\_Start--  Starting point(X Axis- starting) of other rectangle

//   lenB\_end --   Ending point(X Axis- starting) of other rectangle

//   HeightA\_start -- Starting point(Y Axis- starting), of first rectangle

//   HeightA\_end  --  Ending point(Y Axis- starting), of first rectangle

//   HeightB\_Start--  Starting point(Y Axis- starting) of other rectangle

//   HeightB\_end --   Ending point(Y Axis- starting) of other rectangle

//   lengthoverlap\_start --- point from where 2 rectangle sides are parallel - X Axis(length)

//   lengthoverlapcount --- Number of points 2 rectangle sides are parallel - X Axis(length)

//   Heightoverlap\_Start --- Point from where 2 rectangles are in parallel - Y Axis(breadth)

//   Heightoverlapcount --- Number of points where 2 rectangles are in parallel - Y Axis(breadth)

//------------------------------------------------------------------------------

int main()

{

        int lenA\_Start, lenB\_Start , lenA\_End , lenB\_End ,lengthoverlapcount =-1, lengthoverlap\_Start;

        int HeightA\_Start, HeightB\_Start , HeightA\_End , HeightB\_End , Heightoverlapcount =-1, Heighthoverlap\_Start;

        int left\_x,left\_y,right\_x,right\_y ;

//  Read co-ordinates of first rectangle

        printf("\n Enter co-ordinates of first rectangle, Top left corner cordinates followed by bottom right coordinates\n" ) ;

        scanf("%d%d%d%d",&left\_x, &left\_y , &right\_x,&right\_y);

//Convert input co-ordinates in to starting point(X/Y axis) and length

         lenA\_Start = left\_x ;

         HeightA\_End = left\_y ;

         lenA\_End = right\_x ;

         HeightA\_Start = right\_y ;

//  Read co-ordinates of second rectangle

        printf("\n Enter co-ordinates of second rectangle, Top left corner cordinates followed by bottom right coordinates\n" ) ;

        scanf("%d%d%d%d",&left\_x, &left\_y , &right\_x,&right\_y);

//Convert input co-ordinates in to starting point(X/Y axis) and length of the side

         lenB\_Start = left\_x ;

         HeightB\_End = left\_y ;

         lenB\_End = right\_x ;

         HeightB\_Start = right\_y ;

        if (lenA\_Start < lenB\_Start)

        {

                if(lenA\_End == lenB\_Start)  // if one rectangle end point(X-Axis) becomes starting point for other rectangle

                {

                        printf("\n no intersection found\n ");

                        return 0;

                }

                else

                {

                        intersection(lenA\_Start,lenA\_End,lenB\_Start,lenB\_End,&lengthoverlap\_Start,&lengthoverlapcount) ;

                }

        }

 else

        {

                if(lenA\_End == lenB\_Start)      // if one rectangle end point(X-Axis) becomes starting point for other rectangle

                {

                        printf("\n no intersection found\n ");

                        return 0;

                }

                else

                {

                        intersection(lenB\_Start,lenB\_End,lenA\_Start,lenA\_End,&lengthoverlap\_Start,&lengthoverlapcount) ;

                }

        }

        if(lengthoverlapcount >1 ) // Minimum 2 in points are overlapping in horizonal direction(X-Axis)

        {

                if (HeightA\_Start < HeightB\_Start)

                {

                        if(HeightA\_End == HeightB\_Start)// if one rectangle end point(Y-Axis) becomes starting point for other rectangle

                        {

                                printf("\n no intersection found\n ");

                                return 0 ;

                        }

                        else

                        {

                                intersection(HeightA\_Start,HeightA\_End,HeightB\_Start,HeightB\_End,&Heighthoverlap\_Start,&Heightoverlapcount) ;

                        }

                }

                else

                {

                        if(HeightB\_End == HeightA\_Start) // if one rectangle end point(Y-Axis) becomes starting point for other rectangle

                        {

                                printf("\n no intersection found\n ");

                                return 0;

                        }

                        else

                        {

                                intersection(HeightB\_Start,HeightB\_End,HeightA\_Start,HeightA\_End,&Heighthoverlap\_Start,&Heightoverlapcount) ;

                        }

                }

                if(Heightoverlapcount > 1)

                {

                        printf("\n Intersection is found\n");

                        printf("\n Final Vertices of resultant Rectangle is (%d,%d),(%d,%d),(%d,%d),(%d,%d)",

                                        lengthoverlap\_Start, (Heighthoverlap\_Start + Heightoverlapcount-1),

                                        (lengthoverlap\_Start+lengthoverlapcount-1), (Heighthoverlap\_Start + Heightoverlapcount-1),

                                         lengthoverlap\_Start, Heighthoverlap\_Start ,

 lengthoverlap\_Start+lengthoverlapcount-1 ,Heighthoverlap\_Start  );

                }

                else

                {

                        printf("\n no intersection found\n ");

                }

        }

        else

        {

                printf("\n no intersection found\n ");

        }

        return 0 ;

}

/// \function : intersection

//------------------------------------------------------------------------------

//

///

/// \brief To check whether sides are getting intersected and point of intersection lengths

///

///

//   start -- Starting point(X/Y Axis- starting), could be length/breadth

//   end  --  Ending point(X/Y Axis- starting), could be length/breadth

//   start\_ref -- Starting point(X/Y Axis- starting) of other rectangle, could be length/breadth

//   end\_ref -- Ending point(X/Y Axis- starting) of other rectangle, could be length/breadth

//   overlap\_start -- point(X/Y Axis- starting), could be length/breadth where possible overlap starts

//   \*count-- number of points possible overlap can be seen, point(X/Y Axis- starting), could be length/breadth

//------------------------------------------------------------------------------

void intersection(int start,int end,int ref\_start,int ref\_end,int \*overlap\_Start,int \*count)

{

        \*(count) = 0 ;

       printf("\n Initial Values are %d\t %d\t%d\t%d\n",start,end,ref\_start,ref\_end) ;

       for (;(start<=end) && (start <= ref\_end);) // check till end of side(X/Y Axis) )

        {

           if( start==ref\_start )    // If Two sides are in parallel

           {

                   if(\*count == 0 )

                   {

                       \*overlap\_Start = start ;  //  Acquire overlap start point

                   }

                   start++;

                   ref\_start++;

                   (\*count)++ ;

           }

           else

           {

                start++;

           }

        }

}