REGULATIONS

Due date: 11 December 2017, Monday *(Not subject to postpone)*

Submission: Electronically. You will be submitting your program source code through a text file which you will name as `the2.py` by means of the COW system.

Team: There is **no** teaming up. This is an EXAM.

Cheating: Source(s) and Receiver(s) will receive zero and be subject to disciplinary action.

INTRODUCTION

This exam is about determining properties of line segments and polygons with three or four edges.

Let us call a three or four edged polygon (a triangle or a quadrilateral) as a *minority-shape*. A minority-shape is described by a list of vertices such that the vertices are placed in the list in the order of a *counter-clockwise* traversal of the vertices on the shape. However, no specification of the starting vertex is given, which suggests that, with such a representation, generally speaking, a minority-shape with n vertices could be represented by n different lists. A *vertex* is represented as a tuple of two floating point numbers, i.e., as (x, y), where x and y are the horizontal and vertical coordinates, respectively.

PROBLEM & SPECIFICATIONS

- You are expected to write a function that you will name exactly as `minority_shape_intersect`. This function will take <u>two</u> minority-shapes (as lists of tuples) as argument.

- The expected return value is the representation of the polygon that is formed by the intersection of the two minority-shapes given as argument. Similar to the representation of minority shapes, your return value is a list of vertices, traversed *counter-clockwise*, starting with any vertex.

- In case the intersection is empty, the return value will be an empty list. It is also possible that the intersection is one of the minority-shapes provided as argument to the function.

- Any two floating points x and y will be considered having the same value iff $|x - y| < 0.001$.

EXAMPLE
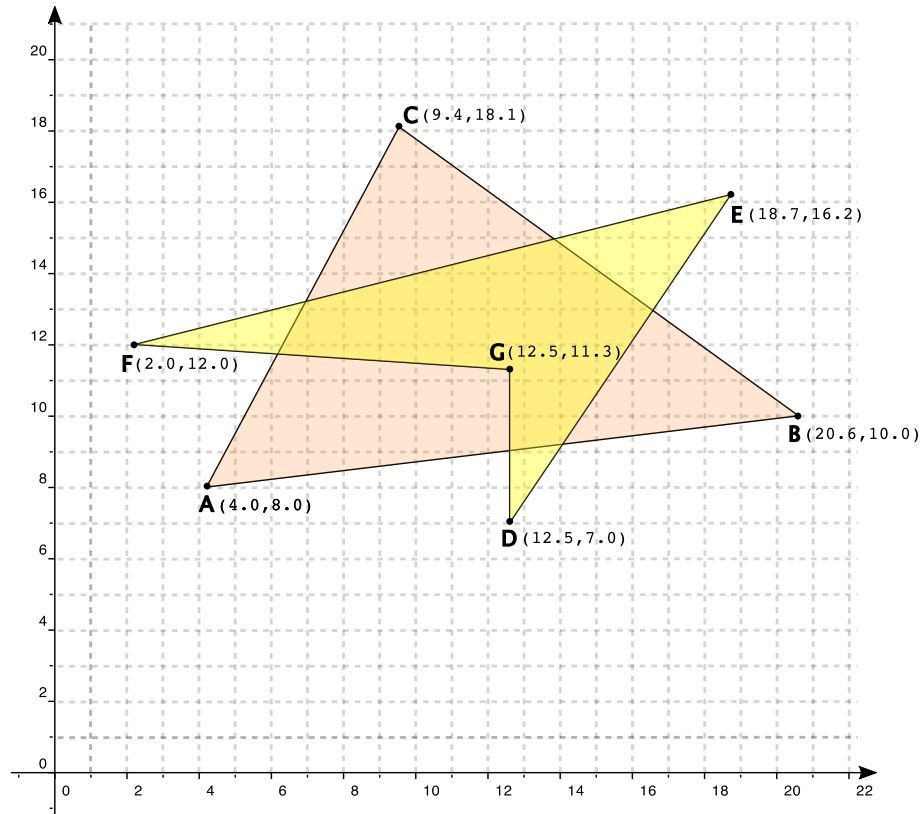

```
>>> minority_shape_intersect([(4.,8.),(20.6,10.),(9.4,18.1)],\
                    [(12.5,7.),(18.7,16.2),(2.,12.),(12.5,11.3)])
[(12.5,11.3), (12.5, 9.024096385542167), (13.984606613454961, 9.202964652223491),
 (16.513454260733393, 12.955448257862459), (13.748900224891674, 14.954813230212277),
 (6.781560380848003, 13.202548119734228), (5.996175908221797, 11.733588272785214)]
```

Your result may differ from the above within an error defined in the previous section. Also any cyclic permutation of the output list would also be valid. You are allowed to define as many helper functions as you like. You cannot use the `import` statement.

NOTES

- Your function will be tested with multiple data.

- Grade-wise 70% of the input data consists of triangle-triangle intersections.

- Any program that performs only 30% and below will enter a glass-box test (eye inspection by the grader TA). The TA will judge an overall THE2 grade in the range of [0,30].

- The glass-box test grade is not open to negotiation, discussion or explanation.

- Your function will be tested with Python interpreter (v2.7) that is installed on *inek* machines running Linux.

- You are encouraged to share input-outputs on the news group of the course.

- No degenerate triangles or degenerate quadrilaterals will be given (No three vertices of a minority-shape are collinear).