FindSafeTriangle.by

This program aims to find the vertices of the triangle with the largest area from a list of supplied coordinates The main program loop consists of the following:

- Reading and parsing input file from the supplied arguments.
- Using the `itertools` module's combination method to generate all possible triable.
- Iterating through the generated triangles, calculating and storing the area for each using the `find_area_tri()` function.
- Obtaining the coordinate combination that would result in a triangle with the largest area by determining the maximum area value from those stored.

FindSafePolygon.py

This program aims to generate convex polygons from smaller adjacent triangles and determine the polygon with the largest area from a list of supplies coordinates The main program loop consists of:

- Reading and parsing an input file from the supplied arguments
- Using the `combination` method to generate a list of all possible triangles and polygons
- Finding the area of the generated polygons and triangles using the `find_poly_area` and `find_area_tri` functions, respectively.
- Only calculating the area of a polygon if it is deemed to be convex, checked by the `is_convex` function
- Recording the maximum polygon and triangle areas for appropriate output

FindSafeSurrounding.py

This program aims to triangulate an earthquake survivor's possible location. If triangulation is not possible, the closest points to them are reported. The main program loop consists of:

- Reading and Parsing an input file from the supplied arguments
- Isolating the survivor's location.
- Determining possible locations by generating triangles.
- Determining whether the survivor or other obstructions are in the area identified by triangulation using the `is_inside_tri` function

`find_length(coord1, coord2)`

This function calculates the distance between two coordinates using the formula.

$$\text{Length } = \sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2}$$

`find_area_tri(coords1, coords2, coords3)`

Used to calculate the area of the triangle using Heron's Forumula, using the length of the triangle's sides (obtained with the `find_length` function).

$$\text{Area } = \sqrt{s(s-a)(s-b)(s-c)}$$

Where a, b and c are the lengths of the sides and s is half the perimeter of the triangle, i.e. $\frac{(a+b+c)}{2}$

`find_poly_area(coords1, coords2, coords3, coords4)`

Calculates the area of a polygon using the Shoelace Formula

$$\text{Area } = \frac{1}{2}\left|\begin{matrix} x_1 & x_2 & \dots & x_n & x_1 \\ y_1 & y_2 & \dots & y_n & y_1 \end{matrix}\right|$$

`is_convex(polygon)`

Checks if a polygon is convex by calculating the cross product of three consecutive points in the polygon. If the cross product is positive, it is convex; otherwise, it is concave.

`is_inside_tri(point, triangle)`

Checks if a point is inside a triangle by calculating the area of the triangle and the sum of the areas of the three triangles formed by the point and the triangle's three vertices. If the sum of the areas is equal to the area of the triangle, the point is inside the triangle, otherwise, it is outside.